

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-142830

(43)Date of publication of application : 25.05.2001

(51)Int.Cl.

G06F 13/00
H04L 12/18

(21)Application number : 2000-270931

(71)Applicant : FUJI XEROX CO LTD

(22)Date of filing : 07.09.2000

(72)Inventor : ELIZABETH F CHURCHILL
LESTER D NELSON
SALA BLY
CATHERINE C MARSHALL
JOHNATHAN TREVOR
SULLIVAN JOSEPH W

(30)Priority

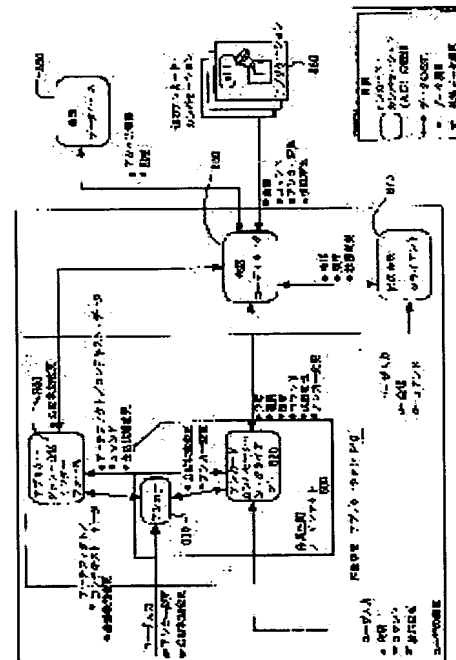
Priority number : 1999 391069 Priority date : 07.09.1999 Priority country : US

(54) ANCHORED CONVERSATION SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a means for anchoring a communication client at any arbitrary position inside an artifact.

SOLUTION: An artifact 800 is accessed in an application 810 and each of anchored conversations(AC) in the artifact is capsuled as an AC client 820. Each of AC clients provides a UI for a user to start a conversation and is related with an anchor 830 for marking the position inside the artifact. An application conversation interface 840 is added to each of applications and performs related application processing, data access, operation and other conversation processing for anchoring the AC client inside the artifact. A conversation coordinator 850 maintains communication between the AC and the other conversation client and accesses a conversation data base 860 containing a conversation history and access control data.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision
of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2001-142830
(P2001-142830A)

(43) 公開日 平成13年5月25日 (2001.5.25)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード* (参考)
G 0 6 F 13/00	6 5 0	G 0 6 F 13/00	6 5 0 B
H 0 4 L 12/18		H 0 4 L 11/18	

審査請求 未請求 請求項の数43 O L (全 41 頁)

(21) 出願番号 特願2000-270931 (P2000-270931)
(22) 出願日 平成12年9月7日 (2000.9.7)
(31) 優先権主張番号 3 9 1 0 6 9
(32) 優先日 平成11年9月7日 (1999.9.7)
(33) 優先権主張国 米国 (U S)

(71) 出願人 000005496
富士ゼロックス株式会社
東京都港区赤坂二丁目17番22号
(72) 発明者 エリザベス エフ. チャーチル
アメリカ合衆国 94110 カリフォルニア
州 サンフランシスコ トウエンティーフ
ィフス ストリート 3740 ナンバー
206
(74) 代理人 100079049
弁理士 中島 淳 (外1名)

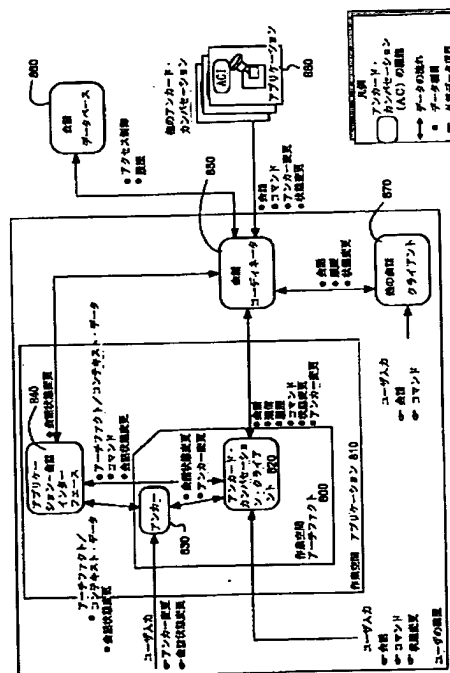
最終頁に続く

(54) 【発明の名称】 アンカード・カンパセーション・システム

(57) 【要約】

【課題】 アーチファクト内の任意の位置に通信クライアントをアンカーする手段を提供する。

【解決手段】 アーチファクト800はアプリケーション810内でアクセスされ、アーチファクト内の各アンカード・カンパセーション (AC) は、ACクライアント820としてカプセル化される。各ACクライアントは、ユーザが会話を開始するためのU Iを提供し、アーチファクト内の位置をマークするアンカー830と関連付けられる。アプリケーション-会話インターフェース840は各アプリケーションに追加され、アーチファクト内にACクライアントをアンカーするための関連アプリケーション処理及びデータ・アクセス及び操作、並びに他の会話処理を行う。会話コーディネータ850は、ACと他の会話クライアントとの間の通信を維持し、会話履歴及びアクセス制御データを含む会話データベース860にアクセスする。



【特許請求の範囲】

【請求項 1】 少なくとも 1 つの会話クライアントをアプリケーションの 1 つ及び該アプリケーションのアーチファクト（産物）にアンカー（固着）するように構成された会話コーディネータ、を含むアンカード・カンバセーション（固着された会話）配置及び維持システム。

【請求項 2】 前記会話コーディネータ及び前記会話クライアントは、前記会話コーディネータが前記会話クライアントの開始及び位置決めを指示するように通信によってリンクされる別個のアプリケーションであることを特徴とする、請求項 1 に記載のシステム。

【請求項 3】 前記会話クライアントにより成される前記通信は、永続的であり、前記会話コーディネータを介してアクセスされることが可能であることを特徴とする、請求項 1 に記載のシステム。

【請求項 4】 前記会話クライアントが携わった会話を格納するように構成されたデータベースを更に含み、前記会話コーディネータは更に、ユーザが前記データベースに格納された任意の現在又は以前の会話にアクセスすることを可能にするように構成されていることを特徴とする、請求項 1 に記載のシステム。

【請求項 5】 前記会話コーディネータが、会話要求を要求側の会話クライアントから受信するように構成された要求ハンドラと、前記要求側の会話クライアントのアプリケーション、アーチファクト、及びアンカー位置に等しいアンカー位置、アプリケーション、及びアーチファクトを有する受信側の会話クライアントを開始するように構成されたクライアント・メカニズムと、を含む、請求項 1 に記載のシステム。

【請求項 6】 前記要求ハンドラは、前記会話要求の意図された受信者がアンカード・カンバセーションへの参加を受諾するか否かを識別する受諾メカニズムを含む、請求項 5 に記載のシステム。

【請求項 7】 前記受諾メカニズムが、意図された受信者への会話の全ての参加者を識別する識別装置を含む、請求項 6 に記載のシステム。

【請求項 8】 前記クライアント・メカニズムが、前記要求ハンドラにより受信された会話要求に関連するターゲット・アプリケーションを開始するように構成された開始メカニズムと、前記会話要求に関連するアーチファクトを検索し、該検索されたアーチファクトを前記ターゲット・アプリケーションにロードするように構成された検索メカニズムと、前記要求側の会話クライアントと前記受信側の会話クライアントとの通信リンクを確立するように構成された接続マネージャと、前記受信側の会話クライアントを前記検索されたアーチ

ファクト内の相応のアンカー位置に配置及びアタッチするように構成されたアンカー装置と、を含む、請求項 5 に記載のシステム。

【請求項 9】 前記アンカー装置が更に、前記アンカーの位置が移動される場合に、前記相応のアンカー位置の場所をトラッキングし、前記会話クライアントのユーザ・インターフェースを前記アンカーの所定の近接範囲内に位置決めするように構成される、請求項 8 に記載のシステム。

【請求項 1 0】 前記アンカー装置が、前記会話クライアントの位置に対応するアンカーに関連して維持するように構成されたアタッチメント装置と、前記会話クライアントの位置に対応するアンカーから分離するように構成されたアタッチ解除装置と、分離された会話クライアントを該クライアントの対応するアンカーの位置に再アタッチするように構成された再アタッチメント装置と、を含む、請求項 8 に記載のシステム。

【請求項 1 1】 会話クライアント間の通信情報を格納するように構成されたデータベースと、以前の通信情報を前記要求側の会話クライアントに関連する前記データベースから検索し、該検索された以前の通信情報を少なくとも前記受信側の会話クライアントにロードするように構成された会話検索メカニズムと、を更に含む、請求項 8 に記載のシステム。

【請求項 1 2】 前記接続マネージャは、前記データベースが、前記開始された会話クライアントが参加した全ての会話のログを前記データベースに記録する無言の参加者であるように、前記データベースを前記開始された会話クライアントに接続する接続を確立する、請求項 1 1 に記載のシステム。

【請求項 1 3】 前記会話クライアントの会話情報をデータベースに格納するように構成されたデータベース・メカニズムを更に含む、請求項 1 に記載のシステム。

【請求項 1 4】 前記会話情報が、前記会話クライアントに対応するアンカーの前記アーチファクト内の位置の少なくとも 1 つの履歴、及び前記会話クライアントと別のクライアントとの間の通信情報を含む、請求項 1 3 に記載のシステム。

【請求項 1 5】 少なくとも 1 つのターゲット・アプリケーションを開始し、該ターゲット・アプリケーションにアーチファクトをロードするように構成された API メカニズムと、会話クライアントを、前記アーチファクトと該アーチファクトのアンカーの何れかにアンカーするように構成されたアンカー装置と、を更に含む、請求項 1 に記載のシステム。

【請求項 1 6】 前記 API メカニズムは、少なくとも 1 つのターゲット・アプリケーションの実行をターゲット・アプリケーション・プログラミング・インターフェ

ースを介して指示するように構成された少なくとも1つのアプリケーション制御装置を含む、請求項15に記載のシステム。

【請求項17】 前記会話コーディネータは、前記受信側の会話クライアントに関連する前記アーチファクトがマスター文書と同期でなくなった場合に、前記受信側の会話クライアントのユーザに通知するように構成された同期装置を更に含む、請求項5に記載のシステム。

【請求項18】 前記同期装置は、ユーザがそれによって前記同期でないアーチファクトを更新するために同期処理を起動することができる起動可能な入力メカニズムを含む、請求項17に記載のシステム。

【請求項19】 前記会話コーディネータが、前記アーチファクトを前記マスター文書への変更に基づいて更新するように構成された更新メカニズムを更に含む、請求項17に記載のシステム。

【請求項20】 前記マスター文書は、前記要求側の会話クライアントのアーチファクトである、請求項17に記載のシステム。

【請求項21】 前記会話コーディネータは、前記受信側の会話クライアントの前記アーチファクト内の他のアンカード・カンパセーションを識別し、識別されたアンカードのそれぞれを所定のアクセス/機密設定に基づいてハイライト、グレー、及び不可視の何れか1つの状態にするように構成されたアンカー識別メカニズムを更に含む、請求項5に記載のシステム。

【請求項22】 前記会話コーディネータが、会話クライアントを開始するように構成されたクライアント・メカニズムと、前記会話クライアントを前記アプリケーション及び前記アーチファクトのアンカーの何れかに関連して位置決めするように構成されるアンカー装置、を含む、請求項1に記載のシステム。

【請求項23】 前記会話コーディネータが、前記アプリケーションからメッセージを受信するように構成されたメッセージング装置を更に含む、請求項1に記載のシステム。

【請求項24】 前記会話コーディネータが、前記会話クライアントから通信ネットワークへの通信、及び通信ネットワークから会話クライアントへの通信をルーティングするように構成された通信装置を含む、請求項1に記載のシステム。

【請求項25】 前記通信装置が、通信をIRC（インターネット・リレー・チャット）ネットワークに渡し、且つIRCネットワークから通信を検索するように構成されたIRCクライアントである、請求項24に記載のシステム。

【請求項26】 前記会話コーディネータが、無言の参加者としてリモート・データベースを前記会話クライアントに接続するように構成されたデータベース・メカニ

ズムを含む、請求項1に記載のシステム。

【請求項27】 アンカーをアプリケーションのアーチファクトと共に選択された位置に配置するように構成された配置メカニズムと、前記アンカーに関してユーザにより成されたアクションを認識するように構成されたアクション装置と、前記成されたアクションを実施するように構成された起動ルーチンと、を含む、アンカード・カンパセーション・サポート装置。

【請求項28】 前記アンカーは、インライン形状内のActiveXコンポーネントを含み、前記アクション装置はルーチンのセットを含み、各ルーチンは前記ユーザのアクションの少なくとも1つの対応に関連し、前記ActiveXコンポーネントにより開始される、請求項27に記載の装置。

【請求項29】 前記アクション装置は、前記ユーザのアクションを該ユーザのアクションを実施するために応答するように構成された会話コーディネータに通信するように構成されたアプリケーション・プログラミング・インターフェースの拡張セットを有するアプリケーション会話インターフェースを含む、請求項27に記載の装置。

【請求項30】 アプリケーションのそれぞれのアプリケーション・オペレーションへのアクセス及び操作を可能にするアプリケーション・アクセス・メカニズムを有するアプリケーション会話インターフェース（ACI）と、会話クライアントを識別し、アーチファクト内の選択されたアンカー位置に会話クライアントをアタッチするように構成されたアンカー・メカニズムと、を含む、コンテキストとしてアプリケーションのアーチファクトを有するアンカード・カンパセーション呼び出しシステム。

【請求項31】 アプリケーションから開始要求を受信し、会話クライアントが前記開始要求に関連するアンカー位置にアンカーされる状態を開始し、要求側の会話クライアントと同じアプリケーション、アーチファクト、及びアンカー位置を有する受信側のホストで、アンカード・カンパセーション・クライアントを開始するためのアンカード・カンパセーション要求を送信する、ように構成された会話コーディネータを更に含む、請求項30に記載のシステム。

【請求項32】 マスター・アーチファクトにアンカーされた会話クライアントから会話要求を受信するステップと、要求側の前記会話クライアントとアンカード・カンパセーションを開始するステップと、

を含む、アンカード・カンパセーションの開始及び調整方法。

【請求項 33】 前記アンカード・カンパセーションを開始するステップが、
会話への招待状を該会話招待状の意図される受信者に送信するステップと、

前記受信者から招待への返答を受信するステップと、
前記招待への返答が肯定である場合、前記受信者との同期のアンカード・カンパセーションを開始するステップと、

前記招待への返答が肯定ではない場合、前記受信者との非同期のアンカード・カンパセーションを開始するステップと、

を含む、請求項 32 に記載の方法。

【請求項 34】 前記アンカード・カンパセーションを開始するステップが、
前記マスター・アーチファクトに対応するアプリケーションを開始するステップと、

前記マスター・アーチファクトのコピーを検索し、前記呼び出されたアプリケーションにロードするステップと、

受信側の通信クライアントを前記要求側の会話クライアントのアンカー・ポイントに対応するアンカー・ポイントにアンカーするステップと、

を含む、請求項 32 に記載の方法。

【請求項 35】 前記開始されたアンカード・カンパセーションを維持するステップを更に含む、請求項 32 に記載の方法。

【請求項 36】 前記アンカード・カンパセーションを維持するステップが、
前記アンカード・カンパセーションのアンカー位置が別のアプリケーション又はアーチファクトに移動されると、前記アンカード・カンパセーションのコンテキストを変更するステップを含む、
請求項 35 に記載の方法。

【請求項 37】 前記維持するステップが、
前記会話の少なくとも 1 人の参加者による要求及び／又はアクションに基づいて、各会話クライアントを終了することにより前記会話を終了させるステップを含む、
請求項 35 に記載の方法。

【請求項 38】 アプリケーションのアーチファクト内にアンカーを配置するステップと、
前記アンカーをアンカード・カンパセーション実行ルーチンのセットにリンクするステップと、
アンカード・カンパセーションを前記アンカーに影響を及ぼすユーザ・アクションに基づいて管理するステップと、

を含む、アンカード・カンパセーション管理方法。

【請求項 39】 前記管理するステップが、前記アンカーに影響を及ぼすユーザ・アクションの少なくとも 1 つ

に基づく前記アンカード・カンパセーション実行ルーチンのセットの少なくとも一部を開始するステップを含む、請求項 38 に記載の方法。

【請求項 40】 前記ユーザ・アクションは、アンカーの呼び出し、アンカーの削除、アンカーの移動の少なくとも 1 つを含む、請求項 39 に記載の方法。

【請求項 41】 前記管理ステップが、
前記ユーザ・アクションにより要求される機能を実施するために外部の会話コーディネータを呼び出すステップを含む、請求項 39 に記載の方法。

【請求項 42】 前記アンカード・カンパセーション実施ルーチンのセットは、前記アプリケーションの API の少なくとも一部である、請求項 38 に記載の方法。

【請求項 43】 アンカード・カンパセーションの位置を識別するアンカーを伴うアーチファクトを有する、要求側ホストにおけるアプリケーションからアンカード・カンパセーション要求を受信するための手段と、

要求側ホストにおける前記アプリケーションに関連する要求側会話クライアントを開始するための手段と、

前記要求側会話クライアントの位置を前記要求側ホストにおける前記アプリケーションの前記アーチファクトの前記アンカーに関連して維持するための手段を含む、前記要求側会話クライアントをアンカーするための手段と、

受信側会話クライアントの開始を要求するための手段と、

受信側ホストで前記アプリケーションを開始するための手段と、

前記アーチファクトのコピーを前記受信側ホストにおいて前記アプリケーション内に引き出す手段と、

前記受信側会話クライアントの位置を前記受信側ホストにおける前記アプリケーションの前記アーチファクトの前記アンカーに関連して維持するための手段を含む、前記受信側会話クライアントをアンカーするための手段と、

を含む、要求側ホスト及び受信側ホストのそれぞれで会話クライアントを開始し、維持するための手段を含む、アンカード・カンパセーション用システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 [著作権表示] 本特許文書の開示の一部には、著作権を保護されるべき資料が含まれる。著作権の所有者は、特許・商標事務所 (PTO) のファイル又は記録に記載されている特許文書又は特許開示が誰かにより複製されても異議はないが、その他の点では如何なるものであれ全ての著作権は当方に所有されている。本発明は、合同及び／又は共同作業環境における通信及び情報の共有化に関する。より詳細には、本発明は、作業のアーチファクト (産物) 又はアプリケーションにアンカー (固着) されたチャット・ボックス・タ

10

20

30

40

50

イブの通信装置の使用に関する。本発明は更に、取り外し可能であり、異なるアプリケーション間で移動可能であり、且つスタンド・アロン・オペレーションが可能なアンカード・カンパセーション (anchored conversation: 固着された会話) に関する。また本発明は、会話のログ及び各会話のアンカー履歴を保持するデータベースを含む、様々なアプリケーションにより開始されるアンカード・カンパセーションの管理に関する。

【0002】

【従来の技術】近年、同じ場所に位置しない個人間での同期及び非同期通信をサポートする多数の使い易いシステムが利用できるようになった。例えば、チャット、IRC、インスタント・メッセージ・アプリケーション (例えば、AOL インスタント・メッセージ)、MUD、及びMOOである (Evard, 1993年; Curtis及びNichols, 1993年)。これらのシステムは、職場でよく使用されていることが証明されており、多数のグループウェアのプラットフォームが今日では、同期チャット・アプリケーション (例えば、ロータスのドミノ (<http://www.software.ibm.com/>)) をサポートする機能を含んでいる。

【0003】このようなシステムは、以下に挙げる点をサポートしている。

- 地理的距離にもかかわらず、迅速且つ少ない労力しか要さないコンタクト
- 他とのコンタクトの割込み式ではない開始
- 進行中及び頻繁な対話の実現性
- 複数の人との通信
- デスクトップ上の別のウィンドウでの異なる個人／グループとの複数の同時会話の実現性

【0004】IRCを使用する実施例は、以下の特徴を有する。IRCは、テキスト・ベースのチャット・クライアントのための単純なプロトコルをサポートするチャット・サーバ・プログラムのネットワークである。テキスト・キーワード／コマンドによりトリガされる、クライアント内における音及びグラフィックのためのアド・オン (add-ons) が存在する。多数のオペレーティング・システム (Windows (登録商標)、Unix、Mac) 用のIRCクライアントが入手可能である (例えば、mIRC)。IRCクライアント・プログラムを使用して、ユーザはテキスト・メッセージを相互にやり取りすることができる。チャット・セッションにログインすると、ユーザは他のチャット参加者に送信するメッセージをタイプ入力することにより「会話する」ことができる。各IRCサーバは仮想的な会議場所を提供する。IRC上で、ユーザは他の人と「チャンネル」(部屋、仮想の場所) で会い、グループで、又は個人的に会話する。所定の討議に参加できる人数、又はIRC上で成立可能なチャンネル数の制限はない。全てのサーバは相互接続されており、IRCネットワークを介してメッセー

ジをユーザからユーザに渡す。1つのサーバは、複数の他のサーバ及び数100台までのクライアントに接続されることが可能である。

【0005】あるMUD及びMOOの例では、会話は持続的な面も有するので、それらはキャッチアップ (遅れた分の取り返し) 及びレビュー (見直し) のためにログを使用することを可能にする。しかしながら、多くのツールの共通の問題は、それらのツールが容易な通信を可能にする一方で、それらは作業のアーチファクトと通信との密接な統合をサポートしないことである。

【0006】会話は、共有されるアーチファクトとは分離した、且つ共有されるアーチファクトには関連又は関係しないウィンドウで行われる傾向がある。これは、図1に例示される。図1では、MUDチャット・ウィンドウ110を含む多数のオープン・ウィンドウが、コンピュータ・ディスプレイ100上に見られる。チャット・ウィンドウ内の会話は通信情報のコンテキストを共有するが、彼らは検討されている作業アーチファクト (例えば、Wordのファイル) のビュー (外観) を共有しない。作業中の文書は別のウィンドウに現われる。

【0007】この分離は、ある種の共同作業、特に「密接に」連携していると特徴づけられる共同作業にとって問題である。密接に連携している共同作業はしばしば、共有されるアーチファクト (例えば、ホワイトボード、モデル、文書等) に関して、会話を通じて他の人と問題を解決することを含む。

【0008】密接に連携している共同作業における会話は、「オブジェクトを載せている (object laden)」 (Fleming, 1998年) と特徴づけられており、検討されている及び／又は共同して制作されているオブジェクト又はアーチファクトに強く焦点があてられる。従って、このような会話は、共有されるコンテキストでは、即ち、全ての共同作業者が、制作されている又は検討されているアーチファクト又はオブジェクトに視覚的にアクセスできる場合にうまく作用する。Flemingの用語では、「オブジェクトが先行し、言葉はそれに続く (object leads and language follows)」。この連続体の他方の末端では、疎連携の協力がしばしば、入念な会話を含む。入念な会話は比較的、実世界からは独立しており、ナレーション、議論及びアプリケーションのインスタンスを含む。ここでは、「言葉が先行し、オブジェクトはそれに続く (language leads and the object follows)」。そして、このようなディスコース (論説) において、オブジェクトはかなり安定したエンティティ (実体／構成要素) である。

【0009】チャット・アプリケーション及びテキスト・ベースの仮想環境を使用している共同作業者ととのインタビューからの証言は、綿密な会話は容易に達成されるが、「オブジェクトを載せている」会話は容易に達成されないことを示している。後者は、ユーザが共有される

チャット・ウィンドウにテキストをペーストするか、又はユーザが会話をしたい特定の内容を共有するために何か別の手段を見出すことを必要とする (Churchill及びBly, 1999年)。

【0010】このようなペースト作業は、ペーストされた資料をそのコンテキストから取り出すという副作用を有する。例えば、文書から取り出されて、チャット空間にペーストされた段落は、もはや文書のその他のコンテキストでは議論されることができない。従って、文書のその他の箇所への言及は、共有される文書無しに成される。

【0011】別の選択肢は、アーチファクト全体を仮想環境に再表示し、アーチファクト及びユーザ (化身の形で) の両方を仮想世界に配置することである。これは、ザ・パレス (the Palace) (<http://www.thepalace.com>) のような2次元のグラフィカルMUDや、人々及びアーチファクトが化身及び共有オブジェクトとして表現されるMASSIVE環境等の3次元の仮想環境において用いられる手法である。

【0012】例えば、図2において、3次元のグラフィックの世界が、共有されるブラックボード200及び化身「ブロック」(210)としてデジタルで再表現されている会話参加者を含んで示されている。単純な2次元のグラフィカル仮想環境 (例えば、ザ・パレス) 内では、バックグラウンド画像が、共有されるコンテキストを提供する。タスクに関するバックグラウンドは、仮想会議が催され得る前に生成される。会話空間の設定及びこの空間への必要なオブジェクトの再現は、時間の面で犠牲が大きく、迅速且つ容易なコンタクト、通信、及び共同作業への妨げを表わしている。

【0013】

【発明が解決しようとする課題】本発明の発明者は、会話 (例えば、チャット空間及び他の通信クライアント等) をアプリケーション及びアプリケーションのアーチファクト内のある位置にアンカーする (一般に、共同開発環境におけるコミュニケーションを高める) ユーティリティを実現させた。従って、ある通信のためのバックグラウンドとしてのアーチファクトと関連して通信が開始され維持されることが可能なように、多様なアプリケーションのアーチファクト内の任意の位置に通信クライアントをアンカーすることが本発明の目的である。

【0014】また、通信クライアントのユーザ・インターフェースの柔軟な配置を可能にすることが本発明の別の目的であり、例えば、参照アーチファクトに関連してアンカーされるか、切り離されて、このアーチファクト内の別の位置に移動されるか、第2のアーチファクトにアタッチされる (取付けられる) か、第2又は更に別のアプリケーションにアタッチされるか、又は任意のアプリケーション又はアーチファクトから切り離されることを可能にする。

【0015】更に、リモート・ユーザの所在位置で、アプリケーションと通信クライアントがアンカーされているアーチファクトと共に通信クライアントを呼び出すために、通信クライアントで始動される通信要求を可能にすることが本発明の別の目的である。

【0016】また更に、会話が配置されるべき位置を参照するアンカーに関する会話のデータベースを保持することが本発明の別の目的である。このデータベースは、会話の履歴及びその会話がどこにアンカーされたかの履歴を含む。

【0017】

【課題を解決するための手段】本発明は、アンカード・カンパセーションの配置及び維持のためのシステムであって、会話クライアントをアプリケーション及びこのアプリケーションのアーチファクトの何れかにアンカーするように構成された会話コーディネータを含む。会話クライアント自体は、任意の通信メカニズムで実施される。

【0018】また本発明は、コンテキストとしてアプリケーションのアーチファクトを有するアンカード・カンパセーションを呼び出すためのシステムでも実施され、このシステムは、各アプリケーションのオペレーションへのアクセス及び操作を可能にするアプリケーション・アクセス・メカニズムを有するアプリケーション会話インターフェース (ACI) と、会話クライアントを識別し、この会話クライアントをアーチファクト内の選択されたアンカー位置にアタッチするように構成されたアンカー・メカニズムとを含む。

【0019】また本発明は、アンカード・カンパセーションを開始し、調整するための方法でも実施され、この方法は、マスター (基本の) アーチファクト内にアンカーされた会話クライアントからの会話要求を受信するステップと、要求側の会話クライアントとのアンカード・カンパセーションを開始するステップとを含む。

【0020】本発明の第1の態様は、少なくとも1つの会話クライアントをアプリケーションの1つ及び該アプリケーションのアーチファクト (産物) にアンカー (固着) するように構成された会話コーディネータ、を含むアンカード・カンパセーション (固着された会話) 配置及び維持システムを提供する。

【0021】本発明の第2の態様は、第1の態様において、会話コーディネータが会話クライアントの開始及び位置決めを指示するように、会話コーディネータ及び会話クライアントが通信によってリンクされる別個のアプリケーションであることを特徴とする。

【0022】本発明の第3の態様は、第1の態様において、会話クライアントにより成される通信が、永続的であり、会話コーディネータを介してアクセスされることが可能であることを特徴とする。

【0023】本発明の第4の態様は、第1の態様におい

て、会話クライアントが携わった会話を格納するように構成されたデータベースを更に含み、会話コーディネータは更に、ユーザがデータベースに格納された任意の現在又は以前の会話にアクセスすることを可能にするように構成されていることを特徴とする。

【0024】本発明の第5の態様は、第1の態様において、会話コーディネータが、会話要求を要求側の会話クライアントから受信するように構成された要求ハンドラと、要求側の会話クライアントのアプリケーション、アーチファクト、及びアンカー位置に等しいアンカー位置、アプリケーション、及びアーチファクトを有する受信側の会話クライアントを開始するように構成されたクライアント・メカニズムと、を含む。

【0025】本発明の第6の態様は、第5の態様において、要求ハンドラが、会話要求の意図された受信者がアンカード・カンパセーションへの参加を受諾するか否かを識別する受諾メカニズムを含む。

【0026】本発明の第7の態様は、第6の態様において、受諾メカニズムが、意図された受信者への会話の全ての参加者を識別する識別装置を含む。

【0027】本発明の第8の態様は、第5の態様において、クライアント・メカニズムが、要求ハンドラにより受信された会話要求に関連するターゲット・アプリケーションを開始するように構成された開始メカニズムと、会話要求に関連するアーチファクトを検索し、この検索されたアーチファクトをターゲット・アプリケーションにロードするように構成された検索メカニズムと、要求側の会話クライアントと受信側の会話クライアントとの通信リンクを確立するように構成された接続マネージャと、受信側の会話クライアントを検索されたアーチファクト内の相応のアンカー位置に配置及びアタッチするように構成されたアンカー装置と、を含む。

【0028】本発明の第9の態様は、第8の態様において、アンカー装置が更に、アンカーの位置が移動される場合に、相応のアンカー位置の場所をトラッキングし、会話クライアントのユーザ・インターフェースをアンカーの所定の近接範囲内に位置決めするように構成される。

【0029】本発明の第10の態様は、第8の態様において、アンカー装置が、会話クライアントの位置に対応するアンカーに関連して維持するように構成されたアタッチメント装置と、会話クライアントの位置に対応するアンカーから分離するように構成されたアタッチ解除装置と、分離された会話クライアントをこのクライアントの対応するアンカーの位置に再アタッチするように構成された再アタッチメント装置と、を含む。

【0030】本発明の第11の態様は、第8の態様において、会話クライアント間の通信情報を格納するように構成されたデータベースと、以前の通信情報を要求側の会話クライアントに関連するデータベースから検索し、

この検索された以前の通信情報を少なくとも受信側の会話クライアントにロードするように構成された会話検索メカニズムと、を更に含む。

【0031】本発明の第12の態様は、第11の態様において、接続マネージャは、データベースが、開始された会話クライアントが参加した全ての会話のログをデータベースに記録する無言の参加者であるように、データベースを開始された会話クライアントに接続する接続を確立することを特徴とする。

【0032】本発明の第13の態様は、第1の態様において、会話クライアントの会話情報をデータベースに格納するように構成されたデータベース・メカニズムを更に含む。

【0033】本発明の第14の態様は、第13の態様において、会話情報が、会話クライアントに対応するアンカーのアーチファクト内の位置の少なくとも1つの履歴、及び会話クライアントと別のクライアントとの間の通信情報を含む。

【0034】本発明の第15の態様は、第1の態様において、少なくとも1つのターゲット・アプリケーションを開始し、このターゲット・アプリケーションにアーチファクトをロードするように構成されたAPIメカニズムと、会話クライアントを、アーチファクトとこのアーチファクトのアンカーの何れかにアンカーするように構成されたアンカー装置と、を更に含む。

【0035】本発明の第16の態様は、第15の態様において、APIメカニズムが、少なくとも1つのターゲット・アプリケーションの実行をターゲット・アプリケーション・プログラミング・インターフェースを介して指示するように構成された少なくとも1つのアプリケーション制御装置を含む。

【0036】本発明の第17の態様は、第5の態様において、会話コーディネータが、受信側の会話クライアントに関連するアーチファクトがマスター文書と同期でなくなった場合に、受信側の会話クライアントのユーザに通知するように構成された同期装置を更に含む。

【0037】本発明の第18の態様は、第17の態様において、同期装置が、ユーザがそれによって同期でないアーチファクトを更新するために同期処理を起動することができる起動可能な入力メカニズムを含む。

【0038】本発明の第19の態様は、第17の態様において、会話コーディネータが、アーチファクトをマスター文書への変更に基づいて更新するように構成された更新メカニズムを更に含む。

【0039】本発明の第20の態様は、第17の態様において、マスター文書が、要求側の会話クライアントのアーチファクトであることを特徴とする。

【0040】本発明の第21の態様は、第5の態様において、会話コーディネータが、受信側の会話クライアントのアーチファクト内の他のアンカード・カンパセーシ

10

20

30

40

50

ョンを識別し、識別されたアンカーのそれぞれを所定のアクセス/機密設定に基づいてハイライト、グレー、及び不可視の何れか1つの状態にするように構成されたアンカー識別メカニズムを更に含む。

【0041】本発明の第22の態様は、第1の態様において、会話コーディネータが、会話クライアントを開始するように構成されたクライアント・メカニズムと、会話クライアントをアプリケーション及びアーチファクトのアンカーの何れかに関連して位置決めするように構成されるアンカー装置、を含む。

【0042】本発明の第23の態様は、第1の態様において、会話コーディネータが、アプリケーションからメッセージを受信するように構成されたメッセージング装置を更に含む。

【0043】本発明の第24の態様は、第23の態様において、メッセージが、アンカー位置、アンカード・カンパセーションの呼び出し、及びアンカーの追加の少なくとも1つを含むことを特徴とする。

【0044】本発明の第25の態様は、第1の態様において、会話コーディネータが、会話クライアントから通信ネットワークへの通信、及び通信ネットワークから会話クライアントへの通信をルーティングするように構成された通信装置を含む。

【0045】本発明の第26の態様は、第25の態様において、通信装置が、通信をIRC（インターネット・リレー・チャット）ネットワークに渡し、且つIRCネットワークから通信を検索するように構成されたIRCクライアントであることを特徴とする。

【0046】本発明の第27の態様は、第1の態様において、会話コーディネータが、無言の参加者としてリモート・データベースを会話クライアントに接続するように構成されたデータベース・メカニズムを含む。

【0047】本発明の第28の態様は、アンカーをアプリケーションのアーチファクトと共に選択された位置に配置するように構成された配置メカニズムと、アンカーに関してユーザにより成されたアクションを認識するように構成されたアクション装置と、成されたアクションを実施するように構成された起動ルーチンと、を含む、アンカード・カンパセーション・サポート装置を提供する。

【0048】本発明の第29の態様は、第28の態様において、アンカーが、インライン形状内のActiveXコンポーネントを含み、アクション装置はルーチンのセットを含み、各ルーチンはユーザのアクションの少なくとも1つの対応に関連し、ActiveXコンポーネントにより開始されることを特徴とする。

【0049】本発明の第30の態様は、第29の態様において、ルーチンが、アプリケーション内で定義されるマクロであることを特徴とする。

【0050】本発明の第31の態様は、第28の態様に

において、アクションが、アンカーの追加、アンカーの移動、アンカーの削除、及びアンカーの起動の少なくとも1つを含むことを特徴とする。

【0051】本発明の第32の態様は、第28の態様において、アクション装置が、前記ユーザのアクションをこのユーザのアクションを実施するために応答するように構成された会話コーディネータに通信するように構成されたアプリケーション・プログラミング・インターフェースの拡張セットを有するアプリケーション会話インターフェースを含む。

【0052】本発明の第33の態様は、アプリケーションのそれぞれのアプリケーション・オペレーションへのアクセス及び操作を可能にするアプリケーション・アクセス・メカニズムを有するアプリケーション会話インターフェース（ACI）と、会話クライアントを識別し、アーチファクト内の選択されたアンカー位置に会話クライアントをアタッチするように構成されたアンカー・メカニズムと、を含む、コンテキストとしてアプリケーションのアーチファクトを有するアンカード・カンパセーション呼び出しシステムを提供する。

【0053】本発明の第34の態様は、第33の態様において、会話クライアントが、テキスト、音、グラフィック、触覚性、ビデオ、及び他の通信媒体の任意の1つを使用可能な通信クライアントを含むことを特徴とする。

【0054】本発明の第35の態様は、第33の態様において、アプリケーションから開始要求を受信し、会話クライアントが開始要求に関連するアンカー位置にアンカーされる状態を開始し、要求側の会話クライアントと同じアプリケーション、アーチファクト、及びアンカー位置を有する受信側のホストで、アンカード・カンパセーション・クライアントを開始するためのアンカード・カンパセーション要求を送信する、ように構成された会話コーディネータを更に含む。

【0055】本発明の第36の態様は、マスター・アーチファクトにアンカーされた会話クライアントから会話要求を受信するステップと、要求側の前記会話クライアントとアンカード・カンパセーションを開始するステップと、を含む、アンカード・カンパセーションの開始及び調整方法を提供する。

【0056】本発明の第37の態様は、第36の態様において、アンカード・カンパセーションを開始するステップが、会話への招待状を該会話招待状の意図される受信者に送信するステップと、受信者から招待への返答を受信するステップと、この招待への返答が肯定である場合、受信者との同期のアンカード・カンパセーションを開始するステップと、招待への返答が肯定ではない場合、受信者との非同期のアンカード・カンパセーションを開始するステップと、を含む。

【0057】本発明の第38の態様は、第36の態様に

10

20

30

40

50

において、アンカード・カンパセーションを開始するステップが、マスター・アーチファクトに対応するアプリケーションを開始するステップと、マスター・アーチファクトのコピーを検索し、呼び出されたアプリケーションにロードするステップと、受信側の通信クライアントを要求側の会話クライアントのアンカー・ポイントに対応するアンカー・ポイントにアンカーするステップと、を含む。

【0058】本発明の第39の態様は、第36の態様において、開始されたアンカード・カンパセーションを維持するステップを更に含む。

【0059】本発明の第40の態様は、第39の態様において、アンカード・カンパセーションのアンカー位置が別のアプリケーション又はアーチファクトに移動されると、アンカード・カンパセーションのコンテキストを変更するステップを含む。

【0060】本発明の第41の態様は、第39の態様において、維持するステップが、会話の少なくとも1人の参加者による要求及び／又はアクションに基づいて、各会話クライアントを終了することにより会話を終了させるステップを含む。

【0061】本発明の第42の態様は、アプリケーションのアーチファクト内にアンカーを配置するステップと、アンカーをアンカード・カンパセーション実行ルーチンのセットにリンクするステップと、アンカード・カンパセーションをアンカーに影響を及ぼすユーザ・アクションに基づいて管理するステップと、を含む、アンカード・カンパセーション管理方法を提供する。

【0062】本発明の第43の態様は、第42の態様において、管理するステップが、アンカーに影響を及ぼすユーザ・アクションの少なくとも1つに基づく前記アンカード・カンパセーション実行ルーチンのセットの少なくとも一部を開始するステップを含む。

【0063】本発明の第44の態様は、第43の態様において、ユーザ・アクションが、アンカーの呼び出し、アンカーの削除、アンカーの移動の少なくとも1つを含むことを特徴とする。

【0064】本発明の第45の態様は、第43の態様において、管理ステップが、ユーザ・アクションにより要求される機能を実施するために外部の会話コーディネータを呼び出すステップを含む。

【0065】本発明の第46の態様は、第42の態様において、アンカード・カンパセーション実施ルーチンのセットが、アプリケーションのAPIの少なくとも一部であることを特徴とする。

【0066】本発明の第47の態様は、アンカード・カンパセーションの位置を識別するアンカーを伴うアーチファクトを有する、要求側ホストにおけるアプリケーションからアンカード・カンパセーション要求を受信する

ための手段と、要求側ホストにおけるアプリケーションに関連する要求側会話クライアントを開始するための手段と、要求側会話クライアントの位置を要求側ホストにおけるアプリケーションのアーチファクトのアンカーに関連して維持するための手段を含む、要求側会話クライアントをアンカーするための手段と、受信側会話クライアントの開始を要求するための手段と、受信側ホストでアプリケーションを開始するための手段と、アーチファクトのコピーを受信側ホストにおいてアプリケーション内に引き出す手段と、受信側会話クライアントの位置を受信側ホストにおけるアプリケーションのアーチファクトのアンカーに関連して維持するための手段を含む、受信側会話クライアントをアンカーするための手段と、を含む、要求側ホスト及び受信側ホストのそれぞれで会話クライアントを開始し、維持するための手段を含む、アンカード・カンパセーション用システムを提供する。

【0067】本発明は、上述の仮想世界に見られるようなアーチファクト及び環境間の現在の関係とは全く異なる手法を取る。会話する場所である共有仮想空間内に世界を再現するよりも寧ろ、本発明は会話を既存の作業コンテキスト内に配置する。以下の項では、アンカード・カンパセーションと呼ばれるこの新しい手法を具体化する適用例が概説される。

【0068】本発明のより完全な理解及び本発明に付随する利点の多くは、以下の詳細な説明への参照が添付図面に関連して考察されることにより本発明がよりよく理解されると、容易に得られるであろう。

【0069】

【発明の実施の形態】本発明は、文書（例えば、Wordファイル、Powerpointファイル、スプレッドシート（表計算）のファイル、図表によるシミュレーション等）などの作業アーチファクトと連携して同期及び非同期通信を可能にする。本発明は、アンカード・カンパセーションと呼ばれる。

【0070】再度、複数の図を通して同じ参照番号が同一の又は対応する構成要素を指示する図面、特に図3を参照すると、アンカード・カンパセーションの実態模型が例示されている。軽いテキスト・ベースの仮想フォーラム300（アンカード・カンパセーション、アンカード・カンパセーション空間、又は会話クライアント）は、ワープロ文書（アーチファクト）310にアンカーされる。仮想フォーラム300では、3人（ユーザであるCathy、Joe及びElizabeth）が、テキストを用いる会話として、このアンカード・カンパセーション空間で同期的に通信している。各ユーザは、会話に関連する文書（アーチファクト）を見ることができる。この文書（アーチファクト）は、その文書の固有（ネイティブ）のアプリケーションにより視覚化され、会話のコンテキストを提供する。

【0071】アンカード・カンパセーションは、認知的

及び計算处理的に安価な方法で共有される作業コンテキストを生成する。この認知的な使い易さは、多くの形で表わされる。

【0072】第1に、会話を作業アーチファクトに取り入れることにより、アンカード・カンパセーションは、共有されるコンテキストが確立され得る前に、会話参加者及びタスク・アーチファクトが仮想世界に再現されることを必要としない。その代わりとして、アンカード・カンパセーションは、既存の作業コンテキスト内に会話を配置する。ユーザは、共有されるコンテキストを設定する必要はない。

【0073】第2に、アンカード・カンパセーションは、使用コストの点で安価である。アンカード・カンパセーションは、既存の文書にアタッチ（アンカー）されるので、アプリケーションの既存の機能性（例えば、Word文書の異なる表示方法）を利用することが可能である。これは、ユーザの既存の専門知識及び文書アプリケーションの使用法に関する知識を利用することが可能であることを意味する。ユーザは、関連文書を読むために不慣れな状況に置かれず、文書の読み方及び操作方法のモデルは適切なままである。対照的に、作業アーチファクトを再表現するツールは、ユーザがそれらのアーチファクトとの新しい方法による対話の仕方を覚えることを必要とする。

【0074】第3に、アンカード・カンパセーションの使用（法）は、異なるアプリケーションにわたって同様であるので、アンカード・カンパセーションのオーバーヘッドは少ない。従って、アンカード・カンパセーションのスプレッド・シートへの挿入は、アンカード・カンパセーションのWord文書への挿入と同じか、又は非常に類似したアクション（動作）を必要とする。更に、画像がバックグラウンドを提供する（例えば、アドビのPhotoshopにおいて）アンカード・カンパセーションで議論の場を設けることは、同一の会話クライアントが利用される際にスプレッドシートに議論の場を設ける場合と同じユーザ・アクションを必要とする。

【0075】第4に、ユーザは、会話リンク（例えば、別のアーチファクトの一部へのハイパー・リンク）をたどることにより、異なる文書及びアプリケーション間を移動することが可能である。このように会話リンクをたどることにより、バックグラウンドの文書が自動的に開かれ、関連文書を開くために必要とされるユーザによる探索を巧みに回避する。ユーザではなく、会話コーディネーター/マネージャが、異なるアプリケーションの文書内の位置へのナビゲーションを処理する。

【0076】アンカード・カンパセーションは、単純に埋め込まれたチャット・オブジェクトではない。会話は、その会話のための局所的なコンテキストに関する情報を記憶する、文書内の「アンカー」を介して文書に添付される。例えば、このコンテキストはアンカーに最も

近い単語、又はスプレッドシート内のユニークなセル識別子（単数又は複数）であり得る。アンカード・カンパセーションが現在挿入されている位置に関するこのような情報は、その作成日時に関する情報及びそれが以前にアンカーされていた位置と共にデータベースに保持される。

【0077】この情報は、全てのアンカード・カンパセーションのデータベースに格納され、そのデータベースには全ての会話も格納される。ユーザは、例えば、あるアンカード・カンパセーションが異なる場所に位置付けられていたことがあるかを知るために、又は他の問い合わせ（例えば、会話の一部の検出、及びキーワード探索）のために、このデータベースを照会することができる。アンカード・カンパセーション・ウィンドウ内で生じるアクション（例えば、アンカーの移動）及び会話は、このデータベースに保持され、探索、見直し、再利用に利用可能である。

【0078】アンカード・カンパセーションは、軽い通信ツールの実用的な利点と、上述の「オブジェクトを載せている」会話をサポートするための要件と、非同期に加えられるメモ（覚書き）のための要件を兼ね備える。好ましい実施の形態において、アンカード・カンパセーションは以下のような特徴を有する。

●地理的に離れている同僚間での同期及び非同期の両方の通信をサポートする。

●会話をするためにユーザが決定する「locales（ロケール）」又はフォーラムを生成する能力を有する。ユーザは、彼らの通信の必要性を受け入れるためにアンカード・カンパセーションを生成することができる。

●「オブジェクトを載せている」会話をサポートするために共有されるコンテキストを確立する能力を有する。全ての参加者は、討議されている文書の同一部分を見る能力を有する。

●ユーザの通信コンテキストの必要性を受け入れるためにアンカード・カンパセーションを生成する能力を有する。

●アンカード・カンパセーションは文書内の関連位置に文字どおりアンカーされるので、ユーザは自動的に関連位置に「連れて行かれ」、共有し討議される関連コンテキストを探索する必要はない。

●使い易く、極短い設定時間のみを要する。仮想フォーラムは作業空間内に配置されるので、共有される仮想フォーラムに作業アーチファクトを示す必要はない。細分性（示される細部の程度、即ち、例えば単語又は写真等のファイルの中味）は、共有フォルダ及びファイルのレベルで実行される共有作業空間モデル（例えば、TeamWare Workplace (Roseman, 1998年)）とは異なることに留意されたい。アンカード・カンパセーションを使用することにより、作業テキスト自体が共有される。

1つの文書内の、又は種々の文書内においてさえも複

10

20

30

40

50

数の位置での複数のアンカード・カンパセーションがサポートされる。従って、ユーザは、同時に異なる個人又はグループとの複数の会話に携わることができる。

●会話をより適切になり得るコンテキストに移動する能力を有する。

【0079】(アンカード・カンパセーションは、文書のコンテキストに配置されることが可能な一方で、それらはまた、一時的に「取り外され(unhooked)」移動されることが可能である。この場合、それらは他のチャット空間アプリケーションと同様に動き、そこでは会話ウィンドウはその「ホーム」文書又はその文書内の位置とは分離している。ユーザにより閉じられると、アンカード・カンパセーションは、その元の位置に戻る。これは、アンカー位置が格納されており、常に使用可能であるので達成される。また、アンカード・カンパセーションは恒久的に「ディアンカーされる(de-anchored: アンカー状態を解かれる)」ことが可能であり、新しい位置に再度アタッチされるか、又は「無アンカー」状態に保たれ、この場合には、会話コーディネータ/マネージャ等の外部のメカニズムを介してのみアクセスされ得る。)

会話のログを記録し、見直すためのメカニズムを有する。これらのログは、アーチファクトに関連し、且つ会話の保持に関するメタデータ又は情報を含み、これらは文書の作成履歴及び使用履歴を提供することが可能である情報を含み得る。

●他の人にメモを残すメカニズムを有する。これらは、文書に関する非同期、且つ共同作業の注記事項を表わす。

●ユーザ自身のアクティビティ(活動)のコンテキストを提供するために、他の人のアクティビティを知ることが可能にする。これは、共同作業のアクティビティの管理に貢献する。

【0080】図4(A)は、テキスト文書420のマージン内に非常に小さく示される、非活動状態のアンカード・カンパセーション410を示し、アンカード・カンパセーションが挿入されていることを示す。参照符号410と同様の表示又は他の任意の表現が、非活動状態の会話を示すために利用されることが可能であり、これらの表現はまた、そこに会話がアタッチされるアンカーの位置を表わす。参照符号410と同様の表示(及びアンカー)は、作成者がそれに関して議論したい文書内の位置に挿入される。

【0081】参照符号410と同様の表示をクリックすることにより、ユーザは会話ウィンドウを開き、他の同僚間で進行中であり得る仮想会話に参加することができる。図4(B)は、活動状態のアンカード・カンパセーション400を示し、これは非活動状態の会話(そしてこの時点で活動状態になる会話)に関連するアンカー

(プッシュピン405で示される)に対応する位置にア

ンカーされる。図4(A)と比べると、プッシュピン405は図4(B)では異なる位置に見られるが、アンカー位置は、会話クライアントを開始することにより変更されない。活動状態のアンカード・カンパセーションは、チャット・クライアント又は他の通信アプリケーションである。

【0082】全ての会話はログが記録されるので、永続的である(例えば、図10のデータベース860を参照)。従って、もし他の同僚が現在ウィンドウ内で活動的に会話をしていないか、又は同期的に議論に参加できない場合でも、ユーザはその仮想議論フォーラムで先に行われた任意の討議を見直すことができる。

【0083】図4(C)は、切り離されたアンカード・カンパセーション450を示す。前述のように、この仮想議論フォーラムには3人の人が居り、アンカード・カンパセーションが議論されている文書の部分に設定されているので、この3人は全て、議論されている段落の同一のビューにアクセスできる。アンカード・カンパセーション・ウィンドウは、必要に応じて図4(C)に示されるようにテキストの脇に移動されることが可能であるが、その文書又は会話空間が閉じられると、常にアンカー・ポイントに戻る。実施の一形態においては、会話が切り離され、文書が閉じられても、会話(会話クライアント)は活動状態のまま維持され、更なる通信が可能である。或いは、文書を閉じることが、会話クライアントを終了するための信号であってもよい。

【0084】図5乃至7は、アプリケーション505

(例としてMicrosoftのWordが使用されているが、説明されている処理は、当業者により他のワープロ・アプリケーション又は他のアプリケーション、例えばグラフィック・パッケージ又はスプレッドシートで実施されてもよい)のslithey_toves(で始まる)文書500へのアンカード・カンパセーションの挿入を例示する。図5では、文書500は、アンカー(プッシュピンとして示される)520に関連付けられた活動状態のアンカード・カンパセーション510を有する。挿入アンカー・マクロ(例えば、表1のアンカーの追加を参照)は、アンカード・カンパセーションを(例えば、感嘆符及びBander snatchの後の閉じ引用符の間の現在のカーソル位置に)挿入するために実行されている。マクロは、このアンカード・カンパセーションに参加するユーザを選択するように、ユーザを促す(例えば、ポップ・アップ選択ボックス530を参照)。ユーザは、あるユーザを会話に招待するために選択し、OKをクリックする。実施の一形態において、ポップ・アップ選択ボックスは、招待ユーザ友だちリスト(例えば、535)のデフォルトのリストを含む。

【0085】ここで図6を参照すると、招待要求は、招待されたユーザの端末550に招待通知540を表示させる。招待要求は、招待されたユーザに、その招待を受

け入れるか、断るか、又はその詳細を見るための機能 5 6 0 を提供する。また、他の機能（例えば、招待する側の人の写真、又は招待する側の人のランク）が提供されることが可能である。また図 6 は、会話コーディネータのユーザ・インターフェース 5 7 0（例えば、本明細書で論じられる会話コーディネータ 8 5 0）を示し、これは、端末 5 5 0 のユーザが携わるアンカード・カンパセーションに関する文書情報を表示する（この時点では、何も文書情報は表示されていないが、jonathan のチャットへの招待を受諾すると、ユーザ・インターフェース 5 7 0 は、jonathan がそこから会話を要求する文書の種類、チャット、そのコピー、及び IP を表示する）。

【0086】図 7 は、アンカード・カンパセーションへの招待の結果及び davor が受諾したことを示す。活動状態のアンカード・カンパセーション 5 1 0 は依然として表示されており、更なる会話 5 8 0 がアンカー 5 9 0 と共に示されている。この会話 5 8 0 では jonathan 及び davor が Bandersnatch! に関する会話に参加している。davor の端末 5 5 0 では、アプリケーション（この例では Microsoft の Word）が自動的に開始され、slithey_toves の文書 5 0 0 のコピーが自動的に検索されて（幾つか例を挙げると、ダウンロード、ファイル読取り／共有、ftp、又は電子メール等の任意の検索メカニズムを介して検索される）アンカード・カンパセーション挿入のポイントが表示され、その会話のためのバックグラウンドを提供する（図示されないが、jonathan の端末上の文書 5 0 0、追加される会話 5 8 0 及びプッシュピン 5 9 0 と同じように見えるであろう）。

【0087】図 7 はまた、アンカード・カンパセーション・クライアントが切り離し可能であることを示す。追加される会話 5 8 0 はデフォルトの固定位置でプッシュピン 5 9 0 に「アタッチ」（アンカー）される（追加される会話 5 8 0 は、文書 5 0 0 がスクロールされても、又はアプリケーション 5 0 5 が画面に対して移動されても、左上の角がプッシュピン 5 9 0 の位置になるように位置し続けられる）ことに留意されたい。他の任意のアンカー位置が利用され、例えば、左下の角、又は左に 1 インチ等が可能である。各会話クライアント（活動状態のアンカード・カンパセーション 5 1 0 及び追加される会話 5 8 0）は、会話クライアントをそれぞれのアンカーにアタッチ及び／又はそれぞれのアンカーから切り離すためのボタンを含む。追加される会話 5 8 0 は、もしボタン 5 9 1 が押されると、その会話とアンカー 5 9 0 との関係を絶たせ、即ちアンカー 5 9 0 からその会話を切り離し、その会話が別の位置にドラッグされるか移動されることを可能にすることに留意されたい。

【0088】また、活動状態のアンカード・カンパセーション 5 1 0 は対応するアンカー 5 2 0 に関連する位置にアンカーされておらず、その代わりに、アンカーよりも上且つ右に位置していることに留意されたい。ボタン

5 9 2 は、もし押されると、会話クライアント 5 1 0 をアンカー 5 2 0 に再アタッチ（アンカー）する。

【0089】各会話クライアントはまた、この会話クライアントのユーザがクライアントの対応アンカー位置を固定することを可能にするロック・ボタンを有する。例えば、活動状態のアンカード・カンパセーション 5 1 0 は、デフォルト固定位置から切り離され、移動されている。ユーザは、固定位置を会話 5 1 0 の現在の位置に再定義するためにロック・ボタン 5 9 3 を利用することができる。この場合は、会話クライアント 5 1 0 は、文書 5 0 0 がスクロールされるか、又はアプリケーション 5 0 5 が異なる位置に移動されても、アンカー 5 2 0 に対して同じ相対位置に留まり続ける。

【0090】更なる機能（図示せず）は、任意の公知のメカニズム（例えば、プルダウン・メニュー又はボタン）により呼び出され得るリセット・メカニズムである。リセット・メカニズムは、追加されたアンカード・カンパセーション 5 8 0 に関して上述されたデフォルトのアンカー設定のようなデフォルト設定に各アンカーをリセットする。

【0091】図 8 は、1 つの文書内の複数のアンカード・カンパセーションを示しており、この例では、プッシュピン（赤で表示される）及び小さいサムネイル 6 0 0 により表わされている。これらの会話空間の任意の 1 つが、単にサムネイルをクリックすることにより呼び出されることが可能である。このアクションは、ユーザが会話フォーラムに参加し、サムネイルが上述のような会話空間を開くことをもたらす。全ての場合において、ユーザはこの共有されるビューの生成のために如何なる設定もする必要はない。既存の文書（この例では word）は、同僚達が既に取り組んでいるものであり、アプリケーションで提示されるビューは、アンカード・カンパセーションが挿入されている場所の概観を取得するために利用される。

【0092】文書の別のビューが図 9 に示され、ここではテキスト・エディタの概要のビューが提供される。従って、アンカード・カンパセーションが挿入されている段落は、同一の視覚的通知キュー（合図）によりマークされる。実施の一形態において、アンカード・カンパセーションを伴うアーチファクトのユーザは、アーチファクトに現存するアンカーなしにこのアーチファクトを保存する（アンカーを伴わない、アーチファクトのクリーン・コピーを作成する）セーブ機能を新たな機能として呼び出すことができる。

【0093】[アンカード・カンパセーションの使用の詳細]

（アンカード・カンパセーションの生成）アンカード・カンパセーション・アプリケーションは、既存の文書と一体化する。従って、ユーザは、例えば、アプリケーション・インターフェース（例えば、マクロ）コマンドを

使用して、非常に容易にアンカード・カンパセーションを文書に挿入することができる。従って、アプリケーションは、アンカーをこのアプリケーションのアーチファクト内に配置するマクロ（例えば、アンカー・コマンド）を含む内蔵された機能、又は変更されたテンプレート（例えば、Word）を有する。

【0094】このコマンドの発行により、アンカード・カンパセーション・インターフェースが、選択された位置に現われる。一旦挿入されると、同僚リスト（例えば、仲間（buddy）リスト）から選択することにより、又は彼らのコンタクト情報をアンカード・カンパセーションのテキスト・ボックスにタイプ入力することにより、適切な同僚と連絡を取ることが可能である。実施の一形態において、アンカー配置コマンドは、ActiveXコマンドを呼び出し、マクロを開始する。このマクロは、会話クライアントを開始するため、並びに選択された同僚と連絡を取るために、会話コーディネータと連絡を取る。友だちリストは任意の方法で保持されることが可能であり、例えば、AOLのインスタント・メッセージの友だちリスト、IRC等の現行の技術を使用して達成されてもよい。

【0095】（会話要求への応答）アンカード・カンパセーションが生成され、同僚が招待されると、通知が招待された人全てのデスクトップに送信される。受信者は単純なダイアログ・ボックス通知を受信し、これが受諾されると、アンカード・カンパセーション空間と、この会話空間が適切な位置にアンカーされている文書が開く。会話がアンカーされているこの文書は、オリジナルのマスター文書のコピーである。

【0096】（アンカード・カンパセーションを去る）会話が終了すれば、アンカード・カンパセーション・ウィンドウは閉じられることが可能であり、文書内のサムネイル・マーカに戻る。

【0097】（アンカード・カンパセーションの切り離し）アンカード・カンパセーションは、図4（C）に示されるように「アンフックされる（unhooked）」、即ち一時的にそれらの位置から移動されることが可能である。他の位置にアンカーされない（即ち、それらのアンカーがリセットされない）限り、それらの会話は、閉じられると同時に最初に配置されていた場所に戻る。

【0098】（アンカード・カンパセーション内の会話及びアクティビティのログの取得）全ての会話はログを取られ、中央のアンカード・カンパセーション・データベースに格納される。会話空間への出入り等の他の全てのイベントもまた記録される。アクション及びテキスト・チャットを含む全てのイベントには時刻がスタンプされる。ログは、文書、アプリケーション、及び作成者ごとに見直されることが可能であり、テキスト及びタイム・スタンプの両方の探索結果が（例えば、会話コーディネータを介して）発行されることが可能である。このよ

うなログをユーザに戻して表示するための多くの形式が可能であり、例えば、複合ログ又は集合体ログが可能である。

【0099】〔コンテキストの格納、アクティビティの格納及びアンカード・カンパセーション〕

（アンカーのためのコンテキストの格納）アンカーは文書内に配置され、それらの位置に対して局所的なテキスト、図形、又は他の物的要素と関連付けられる。もし、あるアンカーがそのコンテキストを失う（例えば、そのアンカーが関連付けられている局所的な単語、文、段落等が削除される）と、このアンカーは、その文書全体と関連付けられるようになる。もし、その文書自体が削除されると、アンカード・カンパセーションはその後、会話データベースを介してアクセスされる。

【0100】（アクティビティの格納）アンカード・カンパセーションは、アンカード・カンパセーションがタッチされている位置と、アンカード・カンパセーションに対して生じたイベント（例えば、生成、切り離し、移動）とに関するログを伴う。これらは、その文書に関するアクティビティ及び通信に関するメタ・データ（例えば、文書又はアンカーID等の文書に加えられる情報、又は会話を保持するために適切な、会話にタッチされる情報）を表わす。

【0101】会話自体は会話データベースに格納され、ユーザにより明示的に削除されない限りは、照会のために利用できる。実施の一形態において、会話コーディネータ／マネージャは、任意の照会を実行する。

【0102】（使用のシナリオ）作業環境でのアンカード・カンパセーションの使用を例示するために、以下のシナリオが示される。

【0103】（シナリオのバックグラウンド）Belinda、Carleen及びDavidは、あるプロジェクトに取り組む同僚である。彼らの仕事は、地質分野のサイトからデータを集めて討議すること、並びに概要報告書、研究記録及び研究成果提示書を作成することを含む。彼らは月ごとのローテーションで働く。即ち、彼らは1ヶ月働くと、次の1ヶ月は働かない。Belinda及びCarleenは第1のローテーション、即ち、同じ時に働き、Davidは第2のローテーションで働く。

【0104】時間帯（時差）及び地理的場所がBelindaとCarleenを隔てている。CarleenとDavidは、同じ地理的場所で作業するが、同じローテーションではなく、即ち、同じ時には働かない。従って、このグループ内では、同期及び非同期通信の両方が存在し、書込みと読取りとの間の時間差の程度は、Belinda及びCarleenでは数分という短さから、特に、前のローテーションで既に成された判断の遅れを取り戻すことが必要になるDavidの場合では1ヶ月まであり得る。彼らのアクティビティは、表1に示されるように表わすことができる。

【表1】

共同作業者 Belinda、Carleen 及び David の時間的及び地理的作業

設定に基づく通信の可能性

	同位置	異なる位置
同時間	同僚の何れも同時間に同位置にはいない	異なる位置で作業する Belinda と Carleen にとっては「あり得る」。異なる時間帯においてではあるが、彼女らの作業時間はしばしば重なる。他の2人とは異なるローテーションで作業する David にとっては「あり得ない」。David は他の2人の何れとも同期の会話は不可能である。
異なる時間	「あり得る」 Carleen と David は異なる時間ではあるが、地理的に同位置で作業する。	「あり得る」 Belinda と Carleen は異なる時間帯（時差のある地域）に位置しているので、非同期に会話し得る。 David は、彼女らとは常に異なる時間にいるので、Belinda と Carleen の両者と非同期に会話する。

【0105】彼らは、彼らの作業に、スプレッドシートと図式表示のパッケージ、テキスト編集アプリケーション、及び PowerPoint のようなプレゼンテーション用パッケージを使用する。彼らは、これらのアプリケーションの全ての文書で協力する。彼らはしばしば、疎遠携状態で作業することを必要とし、レビューのために互いに文書を送信するが、彼らはまた、これらの文書内のデータの詳細を議論すること、及び文書を共著する工程の間は互いにコメントすることを必要とする。これらの議論は、「オブジェクトを載せている」会話のインスタンスである。彼らの作業環境は、同期及び非同期通信の両方が可能であることを必要とする。

【0106】このような「オブジェクトを載せている」会話のための現行の通信手段には、多数の問題点がある。問題の多くは、アーチファクトが共有されることが可能な細分性に関わる。例えば、電話及びボイス・メールは、進展中の作業アーチファクトに関するメタ情報を提供するのに十分であるが、同僚達は共有されるコンテンツに達するまでにあれこれの作業（文書のコピーをプリントしたり、又は彼らのそれぞれのデスクトップ上でアプリケーションを開いたり）を行う。添付文書を送信するために電子メールを使用することができる。しかしながら、この場合でも、文書内の論議されるべき特定ポイントへのナビゲーションは、例えば、「3ページの第2段落の第3文」のように文書内の位置を明記することによって、個々に明記される。

【0107】しかしながら、共有ウィンドウなしには、共同作業者が厳密に何に注意を向けているかといったフ

ードバックは、推論される以外にはほとんど存在しない。従って、会話で述べられる事柄は、与えられる会話によるポイント又はテキストによるポイントに従って各個人により再生される必要のあるそれらのコンテキストを取り除かれる。ビデオ通信ツールは、多数の問題を提示する。専用の文書カメラを設定せずにテキストの細部を共有することには問題があるので、その道具自体を用いて作業することが難しい。この方法を用いる非同期共有もまた、問題を提示する。DocuShare及びBSCW等の共有ファイル記憶装置は、参照されているテキスト内の特定のポイントへのナビゲーションが電子メール又は電話により別に行われる必要があるという点で、電子メールと同様の問題を被る。従って、共有オブジェクトの細分性は、ファイル及びフォルダのレベルであり、文書内のテキストのレベルではない。AOLのインスタント・メッセンジャー並びにMUD及びMOO等の同期ツールは、テキスト、データ、コード等の共有ウィンドウへのペーストを可能にするが、電子メールの場合と同様に、このアクションは送信されている資料の周囲のコンテキストの損失をもたらす。

【0108】（使用されているアンカード・カンパセーションのシナリオ）Belindaは、Carleenが最新のデータ分析を要約した文書を開く。Belindaは、データにおける曖昧さが、Babarサイトが更なる調査を必要とするか否かに関する問題を提起していることを直ちに悟る。Belindaは、何故Carleenがその曖昧さを指摘していないのかを知りたいと思い、直ちに文書のそのポイントでアンカード・カンパセーションを開始する。Belindaは、文

書内の位置を選択し、クリックし、アンカード・カンパセーションを挿入することによりアンカード・カンパセーションを開始することができる。Belinda、Carleen及びDavidは、作業グループのメンバーとして指定されており、これは会話要求のために必要な全てのアドレス指定要素への迅速なアクセスを提供する。

【0109】Belindaは、Carleen及びDavidのコンタクト（交信）名をタイプ入力する。これは、Belindaが彼らとある問題について議論したいという通知をCarleen及びDavidのそれぞれに送信する。もし彼らがログ・オンして、応答可能であれば、彼らはこの招待を受諾することができる。招待を受諾すると、関連のある文書が送信され、CarleenとDavidの両方のデスクトップ上で関連するアプリケーションが起動され、その文書のアンカード・カンパセーションが配置されている場所が開かれる。アンカード・カンパセーション・ウィンドウは、Belindaが議論を希望する文書内のその位置に配置される。

【0110】この例では、Carleenは彼女のコンピュータに向かっていない（そして、Davidはこの月は勤務するシフトではない）。Belindaは、その文書での作業及びデータ分析を続ける。文書内の更に先で、あるデータがCarleenが示した結果とは異なる結果を示唆するとBelindaが考える、別のポイントをBelindaは見つける。Belindaは再び、アンカード・カンパセーションを挿入する。この時点で、この文書内に2つのアンカード・カンパセーションが存在し、何れもCarleen及びDavidをBelindaとの会話に参加するように招待している。しかしながら、それぞれは、文書内の異なる位置に存在する。

【0111】（アンカード・カンパセーションの同期使用）Belindaは作業を続け、極最近は、Agarサイトに関するデータについて疑問を感じている。Carleenは彼女の机に戻り、Belindaとの会話への参加の招待を受けていることに気付く。Carleenは「受諾」をクリックし、直ちにBelindaが最も最近活動していた（即ち、Agarサイトのデータに関する段落内の）アンカード・カンパセーションに参加する。Belindaは実際には文書内の数カ所でCarleenをBelindaとの会話に参加するよう招待したが、全てのインスタンスは、a) 全て同じ文書に関する、b) 全て同じ参加者のグループを対象とする、という点で類似しているので、1つの招待のみがCarleenのデスクトップ上に現われる（本発明の実施の一形態を説明しているが、他の実施の形態も可能であり、例えば、Carleenが選択的に応答することを可能にする複数の類似の招待を表示させてもよい）。このようなアンカード・カンパセーションのセットは、会話の連結空間を生成する。システムのデフォルトは、新規参加者を最も最近活動状態であったコンテナに配置する。

【0112】Carleenが会話を受諾すると、彼女のデスクトップ上には、そのデータ文書の最も最近活動状態で

あったアンカード・カンパセーションの位置、又は最も最近の要求を伝えた会話の位置が開かれる。他のシナリオも可能であり、例えば、最初の会話要求を開始してもよい。しかしながら、Carleenは、その文書内に更なるアンカード・カンパセーションが存在することを知ることができる。彼女は、彼女が導かれた場所のBelindaのコメントを読み、返答を開始する。すぐにCarleenとBelindaは、Agarサイトに関するデータの解釈について議論を始める。

【0113】CarleenとBelindaが、Agarサイトのデータを要約する方法に関してある同意に達すると、Carleenは文書内の別のアンカード・カンパセーションのためのアンカーをクリックする。Belindaは、Carleenが他の箇所で活動しているのを見て、文書内の別の場所に関する会話を続けるためにCarleenに従うことを選択してもよいし、しなくてもよい。

【0114】（アンカード・カンパセーションの非同期使用）1ヶ月後にDavidが彼の作業ルーチンに戻ると、彼はサイト報告文書に気付く。彼は文書のブラウジングを開始し、その文書の様々な位置にアンカード・カンパセーションがアタッチされていることに気付く。しかしながら、彼がデータ又はその分析に関して何らかの疑問を持つのは、Agarサイトに関する結論に至ってからである。Agarサイトのデータのことを読み、Davidは、何故BelindaとCarleenが去年の12月からのデータを含めなかったのかを疑問に感じる。彼はこのポイントのアンカード・カンパセーションをクリックし、彼女らの会話を先頭の方まで素早くスクロールする。先の12月のデータに関しては、どこにも触れられていない。彼はアンカード・カンパセーション内に「December?」とタイプ入力し、実際にBelindaとCarleenが12月のデータに関して会話をしていた序論部分のアンカード・カンパセーションに移動する。彼は、いつBelindaとCarleenがこの話題について議論したかを確認するために、このアンカード・カンパセーションのためのメタデータを調べる。彼は、この会話が最終的な12月のデータが分析された後に行われたことを知って安心する。

【0115】Davidは彼女らの判断にほぼ満足するが、この文書自体にAgarサイト・データを12月に収集したデータに関連して明示的に示した方がよいのではないかと彼は思うというメモを、そのアンカード・カンパセーションに含めようと決心する。Davidは、BelindaとCarleenの何れとも連絡を取れないことを知っているが、彼女らが次にこの文書にアクセスする際に、文書の会話空間が変更されていることに彼女らが気付くであろうことも知っている。Davidがメモを追加したことを示す通知が、会話要求と同様に、彼女らの会話空間に現われる。

【0116】（技術的詳細）アンカード・カンパセーションは、同期及び非同期通信ロケール（例えば、文書内の位置等の、アクティビティが生じる位置で発生する通

信)である。実施の一形態において、通信媒体は、チャット空間等のテキスト・ベースのメカニズムである。しかしながら、例えば、音及びビデオを含む全ての可能な通信媒体が適用可能である。チャット空間とは異なって、それらの通信媒体は、文書及び他のデジタル作業空間(例えば、ウェブ・ページ)にアンカーされることが可能である。それぞれが分離しているチャット空間とは異なり、アンカード・カンパセーションはまた、互いに、そして他の文書ともリンクされる(例えば、文書リンク)ことが可能である。それらは、会話が行われているか、又は行われそうな場所の周囲の作業コンテキストに関する情報にアクセスできる。

【0117】アンカード・カンパセーション(以下では、アンカード・カンパセーションをACと称す)を実施するための本発明の実施の一形態からの機能及びインターフェースのセットが、図10に示される。作業アーチファクト(例えば、文書)800は、アプリケーション(例えば、ワープロ・プログラム)810内でアクセスされる。このアーチファクト内の各アンカード・カンパセーションは、活動状態の場合は、ACクライアント

(例えば、820)としてカプセル化される。

【0118】各ACクライアントは、ユーザがそれを用いて会話を開始するように対話することができる表現を含み、またアーチファクトのある観点に関する会話がそれにより行われ得るユーザ・インターフェース(例えば、図4(B)の400及び図7の500を参照)を提供する。各クライアントは、このアーチファクト内の位置(例えば、局所的テキスト、局所的図形、局所的オブジェクト、局所的セル)をマークするアンカー(例えば、830)と関連付けられる。

【0119】アプリケーション・会話インターフェース(ACI)840は、アンカード・カンパセーション・システムによりサポートされる各アプリケーションに追加されるか、又はアクセス可能にされる。ACIは、アーチファクト内にACクライアントをアンカーするための関連アプリケーション・オペレーション及びデータへのアクセス及び操作(例えば、周囲のテキストへのアクセス)、並びに他の会話オペレーションに対して責任がある。

【0120】会話コーディネータ850は、アンカード・カンパセーションと他の会話クライアント(例えば、「未アンカー」のチャット空間クライアント)との間の通信を維持することに対して責任がある。また、コーディネータは、会話履歴及びアクセス制御データを含む会話データベース860にアクセスする。この節の残りの部分は、これらの機能間の情報の作用及び流れについて詳しく述べる。

【0121】他の会話クライアント870は、それぞれの会話コーディネータを介して会話コーディネータ850と通信する。アンカード・カンパセーション・クライ

アント820と通信する他のアンカード・カンパセーション及び関連アプリケーションは、ACIアプリケーション880として示される。

【0122】(アンカー)実施の一形態において、アンカー(例えば、830)がアーチファクト内に埋め込まれ、このアーチファクトの位置参照フレーム内の位置をマークする。位置参照フレームは、アプリケーション及び/又はアーチファクトのタイプによって異なり得る

(例えば、順次編成ファイル内では文字位置、又はオペレーティング・システムのウィンドウでは[x, y]座標)。またアンカーは、アーチファクト内の位置を識別する表現及びユーザが対話し得る関連会話クライアントを含む。アンカーのための表現例には、ビットマップ画像(例えば、サムネイル410)、ハイライトされたテキスト、又はテキスト、グラフィック及び音の何らかの組合せが含まれる。アンカー表現は静的であってもよく、また情報が直接ユーザ入力から、又は関連する会話クライアント820から、或いはアプリケーション・クライアント・インターフェース840を介して得られたかに基づいて、値が変化してもよい。ユーザは、会話を開始及び終了するため、アンカーを移動又は削除するため、又はアンカー表現の動的部分を変更するために、アンカー表現(例えば、ポインタ及びメニュー・イベント)と対話することができる。この情報は、ACクライアント又はACIを介して他の分散されたソースから同様に提供されることが可能である。

【0123】アンカーは、限定はされないが、次の属性を含む情報をカプセル化する。

●アンカー表現(例えば、活動状態及び非活動状態の両方のアイコン及び/又はテキスト)

●該当するアーチファクトに関するコンテキスト情報(例えば、位置、周囲の内容)

【0124】またアンカーは、アンカーへのユーザ入力時のオペレーションの実施(例えば、関連する会話クライアント820を開始する)を含む。アンカーの実施は、アプリケーション及び/又はアーチファクトのタイプによって異なり、Javaのアプレット、ActiveXコントロール、又は他のこのような埋め込まれたソフトウェア構造を含む。

【0125】Microsoftのオペレーティング・システム(例えば、Windows 95/NT/98)及びアプリケーション(例えば、Word)によく適した一実施例は、会話のためのアンカーとして開発された小型のActiveXコントロールである。制御の各インスタンスは、特定のACを表わし、その画面の状態(位置、フォーカス、露出等)を判断し、エクスポートすることができる。この情報は、ある会話をアプリケーション及び画面のある領域にアタッチするためにAC構造の他の領域により使用されることが可能である。ActiveXコントロールは、多くのMicrosoft文書(Word, Excel, PowerPoint及びIE4)に容易に配

10

20

30

40

50

置されることが可能であり、文書内の中味／内容の流れの一部として扱われる。従って、その文書自体が変更され編集されると、アンカーはその文書内のその位置に絶対的に維持され、アンカーが元々追加されたテキストと共に移動する。

【0126】実施の一形態において、アンカーは、ワードのインライン形状内のActiveXコンポーネントである。インライン形状は、テキスト、写真又はグラフィック等の範囲であり得る。プッシュピン（例えば、410の一部）を参照すると、この実施の形態において、このプッシュピンは、アンカー構成全体を含むインライン形状内のActiveXコンポーネントのフレーム画像である。ActiveXコンポーネントは、アプリケーション（この例ではWord）とACI（及びアプリケーションのAPI）との間の通信に関するコードのセットである。このプッシュピン上でのダブル・クリックは、イベントをActiveXコンポーネントに渡し、ActiveXコンポーネントは情報（例えば、会話を開始するための情報）をACI（例えば、以下で説明されるWordのマクロ910）内のコード（又はコード・モジュール）の適切な部分に配布する。

【0127】（アンカード・カンパセーション・クライアント及び他の会話クライアント）アーチファクト内の各アンカード・カンパセーションは、アンカード・カンパセーション・クライアントとしてカプセル化される。各ACクライアントはユーザ・インターフェース（UI）を提供し、これによりアーチファクトのある観点に関する会話が行われ得る。ACクライアントは、アーチファクト内に配置されたアンカーと関連付けられる。ACクライアントは、何れのアンカーにも関連付けられなくてもよいし（その文書に関する会話）、1つ又は複数のアーチファクト内に配置された1つのアンカー（議題が1つの会話）、又は複数のアンカー（議題が複数の会話）に関連付けられてもよい。最後の例では、他のアンカー・ポイントがこのACにとって有効であり得るが、1つのアンカーが、その主要な視覚的アタッチメントと考えられ得、即ち、このアンカーはACがホームとするアーチファクト・アンカーである。

【0128】ACクライアントは、活動状態（即ち、会話に参加している）か、非活動状態（例えば、休止中、睡眠中、不在）であり得る。各ACクライアントのUIは、1つ又は複数の他のクライアント（AC又は他のクライアント）との会話がそれにより行われることが可能なメカニズム（例えば、参加者情報又はアドレスを要求するウィンドウへ導く会話ボタン）を提供する。会話UIは、独立したオペレーティング・システム・ウィンドウであっても、アプリケーション内で浮動する埋め込まれたオブジェクトであってもよい。各ACクライアントのUIは、（上述の）アンカーとの対話によってユーザにより呼び出されることが可能である。活動状態の会話のACクライアントのUIの位置は、アンカーの現在の

位置に結びつけられる（例えば、アンカーの極近接位置に維持される）。このアンカーが移動し、視界を変化させると、ACクライアントUIは、対応するように移動し、その視界を変化させる。動作は、アンカーが表示されるウィンドウの動き、及びアンカーがアタッチされている内容の変化（例えば、パン（水平移動）、ズーム、スクロール）を含む。

【0129】ACクライアントUIは、ユーザによる入力操作（例えば、メニュー又はポインタ・イベント）によってアンカーから一時的にディアンカー（又は「アンフック」）されることが可能である。ACクライアントUIはその後、そのアンカーに関連するその他の全ての関係及び情報の流れを維持する一方で、このアンカーから独立して位置付けられることができる。クライアントは、恒久的にディアンカーされてもよい。この場合は、UIはこのアーチファクト全体と関連付けられてもよく（例えば、文頭又は末尾の脚注の一部として）、或いはこのアーチファクトとの関連を解かれてもよい（例えば、自立型の会話クライアントとして）。

【0130】ACクライアント820は、他の会話クライアントとの間で会話構成要素（例えば、テキスト、グラフィック、音又は他のマルチメディア・クリップ又はストリーム）の交換を可能にする。ユーザが任意の会話構成要素を入力すると、ACクライアントは、この構成要素を適切に配布するために会話コーディネータ850の（後述の）機能を使用する。ユーザはまた、コマンドをテキスト・コマンド・ラインの形式で入力することもでき、このコマンドは他のクライアントに配布される。コマンドは、アンカード・カンパセーション特有のオペレーション（例えば、アンカーの表現又は位置の変更、アンカーの追加及び削除、開始等の会話状態の変更）又はアーチファクト／アプリケーション特有のコマンド（例えば、カット、ペースト、フォーマット変更）を含む。各ACクライアントは、他のクライアントからコマンド情報を受信することができ、これらのコマンドの解釈及び関連オペレーションの実行に関して責任がある。これらのコマンドの処理中のエラーは、クライアントにとってエラー報告が使用可能である場合は、会話チャンネルを介して報告が返される。アーチファクト／アプリケーションに特有のコマンドは、ACIの呼び出しを介して実行される。

【0131】ACクライアントは、コンテキスト情報（例えば、現在のアンカー位置、周囲のコンテキスト）の要求、及びアーチファクト／アプリケーション特有のコマンド処理に基づく更新のために、アプリケーション会話インターフェース（ACI）にアクセスする。次にACIは、次の状況でACクライアントにアクセスし得る。外部クライアントによって非活動状態のACクライアントが呼び出されると、ACIは、所定のアーチファクトを呼び出し、ユーザのビューを適切なアンカー・ボ

イントに向け、そして会話を開始するために使用されることが可能である。

【0132】ACクライアントは、他の会話クライアントとの接続を確立する（アクセス制御権の確認を含む）ため、ACクライアントの初期化時又はユーザからの要求時に履歴データを受信するため、及び会話、コマンド、アンカー変更、及び会話状態の変化を通信するために、アプリケーション会話インターフェース（ACI）にアクセスする（但し、インターネット・リレー・チャットに基づくクライアント等の幾つかの実施は、外部位置への直接通信を可能にし得る）。

【0133】アンカード・カンパセーション・クライアントは、限定はされないが、以下の属性を含む情報をカプセル化する。

会話内容及び接続データを含む進行中の会話の情報
会話履歴

会話制御状況（例えば、エラー報告、クライアントがそのアンカーからアンフックされた場合の指示、又は他の状況インジケータ）

アーチファクト及びアンカーに関するコンテキスト情報（例えば、位置、周囲の内容）

【0134】ACクライアントの実施は、アプリケーション及び／又はアーチファクトのタイプにより異なり、独立して実行されるプログラム（例えば、Java又はVisual Basicプログラムを含むオペレーティング・システムのウィンドウ）、又はテキスト上を浮動し得る、埋め込まれたアプリケーション特有の構成要素（例えば、Javaのアプレット、ActiveXコントロール、又は他のこのような埋め込まれたソフトウェア構成）を含む。動的に更新されるように変更されている流動的な文書の注記事項（即ち、コンテキスト内の注記事項）はまた、アンカー及び／又はACクライアントの表現のために使用されるであろう。流動的な文書の注記事項（流動リンク）は、どちらもその全体が参考文献として本明細書に組み入れられる、Zellweger他による『情報を与えられた（インフォームド）及び増分の（インクリメンタル）リンク・トランザクションのための流動リンク（Fluid Links for Informed and Incremental Link Transactions）』（ACM Hypertext '98、1998年、50乃至57ページ）、及びChang他による『流動文書のための折衝構成（Anegotiation architecture for fluid documents）』（UIST '98、Proceedings of the 11th annual ACM symposium on User Interface software and technology、123乃至132ページ）に記載されている。

【0135】会話データベース860、ACクライアント、及びアプリケーションの分離は、アプリケーションにとって最適な手法（例えば、MS Wordに対してはActiveXコントロール、ウェブ・ページに対してはJavaのアプレット）に依存して、ACクライアントをアプリケーションに埋め込むための種々の手法の統合及び共存を可能

にする。この分離は、独立した機能コンポーネントを有する柔軟結合モジュール（例えば、別々ではあるがリンクされている、UDPメッセージ又はhttpインターフェースを介してリンクされている）により達成される。

【0136】会話通信は、インターネット・リレー・チャット（IRC）等のハイ・レベルの通信メカニズム又はJava及び他の言語で提供されるロー・レベルのネットワーク接続（例えば、UDP、TCP/IP）を含む既存のネットワーク規格で実施されることが可能である。

【0137】実施の一形態において、本発明は、アンカード・カンパセーションの実施のためにIRCプロトコルを利用する。ACクライアントは、IRCの不要な機能を隠すために、標準のIRCクライアントよりも非常に簡素なインターフェースを提供し得る（他のプロトコルもまた適用されることが可能であり、例えば、CUSEeMe、AOLのインスタント・メッセンジャー、及びICQが適用されてもよい）。

【0138】ヘルプ、クライアント間プロトコル（CTCP：Client-To-Client Protocol）コマンド、ログ制御等を含む多数のチャット・コマンドがサポートされる。従ってIRCは、アンカード・カンパセーションのための基礎的な通信メカニズムを提供する。各ACクライアントは、ACクライアントのアンカーと関連付けられたアーチファクトの内容に関連する議論を保持するIRCチャンネルにリンクされる。

【0139】IRCサポートに加え、ACクライアントは以下のコンポーネントを含む。

【0140】（コマンド・インタープリタ）コマンド・インタープリタは、ACコマンド・テキストを構文解析し、適切なアクションを実行する。コマンド・インタープリタは、構文解析ルーチン生成システム（例えば、Lex/Yacc）及び／又はコマンド・ライン・ユーティリティ（例えば、Java組込み関数、フリーウェア関数ライブラリ（www.freewareweb.com参照））によりサポートされることが可能である。インタープリタは、アプリケーション／アーチファクトに特有のコマンドのためのACI及び会話特有のコマンドのためのIRCコマンド構文への呼び出しを備えられている。

【0141】（アンカー・トラッカー）アンカー・トラッカーは、ACクライアント・ウィンドウを関連付けられたアンカーの上に配置された位置を追跡する。ACIが、このアンカーの位置を提供し、アンカー・トラッカーはこのクライアント・ウィンドウを移動する。トラッキングは、イベントにより起動されてもよいし、又はその位置のポーリングにより行われてもよい。

【0142】（履歴コンポーネント）履歴コンポーネントは、会話履歴を表示し、会話コーディネータからの履歴の更新を受諾する。

【0143】（UI要素）IRCクライアント・ユーザ

・インターフェース要素に追加されるユーザ・インターフェース要素は、履歴要求、アンカー変更、会話状況の変更（例えば、会話の終了、会話ウィンドウを関連付けられたアンカーからアンフックする）のためのメニュー／ポインタ選択を含む。

【0144】（アプリケーション－会話インターフェース）アンカード・カンバセーション・システムによりサポートされる各アプリケーションは、ACI（例えば、840）により拡張される。ACIは、ACクライアントをアーチファクト内にアンカーするための関連アプリケーション・オペレーション及び関連データへのアクセスと操作に対して、並びに他の会話オペレーションに対して責任がある。ACIの幾つかのオペレーションは、個々のアーチファクトに特有であり（例えば、特定文書内のアンカー・ポイントの列挙）、他のオペレーションは、そのアプリケーションに関連付けられる全てのアーチファクトにわたって適用可能である（例えば、アプリケーション・ウィンドウをディスプレイ画面上に配置する）。

【0145】ACクライアントは、コンテキスト情報（例えば、現在のアンカー位置、周囲のコンテキスト）を要求するため、及びアーチファクト／アプリケーション特有のコマンド処理に基づいて更新を行うためにACIにアクセスする。次にACIは、次の状況でACクライアントにアクセスし得る。外部クライアントによって非活動状態のACクライアントが呼び出されると、ACIは、所定のアーチファクトを呼び出し、ユーザのビューを適切なアンカー・ポイントに向け、そして会話を開始するために使用されることが可能である。

【0146】ACIは、アプリケーション（例えば、word）のアプリケーション・プログラム・インターフェース（API）及びラッパー、即ちAPIの拡張部分を含み得る。ラッパー又は拡張部分は、ある機能を実行するために標準のAPIルーチンを包み込む、即ち利用するプログラムである。ラッパーは、アーチファクト（Word文書）の一部であるか、アプリケーション内のテンプレートであるか、又はアプリケーション自体の一部であり得る。或いは、ACIは、アプリケーションとは別の、これもまた標準APIルーチンを用いるラッパーを有するAPIを利用するプログラムであってよい。本明細書に記載されている本発明の各構成要素と同様に、特定部品の実施方法には多数の代案が存在し、実施の実際の形式は、本明細書に提示される教義の範囲内で、ソフトウェア又はシステム設計者に開かれている。

【0147】ACIは、限定はされないが、以下の属性を含む情報をカプセル化することができる。

アーチファクト内に埋め込まれたACクライアントのリスト

アプリケーション及びアーチファクト・ウィンドウの位置

アンカード・カンバセーションの一部である共有アーチファクトのためのバージョン管理データ

【0148】ACIの実施は、アプリケーション及び／又はアーチファクトのタイプによって異なり、独立して実行されるプログラム（例えば、Java又はVisual Basicプログラムを含むオペレーティング・システムのウィンドウ）、オペレーティング・システムのエージェント（例えば、Microsoft Agents）、又は埋め込まれたアプリケーション特有の要素（例えば、Javaのアプレット、ActiveXコントロール、マクロ定義、スクリプト言語コンポーネント、又は他のこのような埋め込まれたソフトウェア構造）を含む。

【0149】ACIは、以下のコンポーネントを有する。

●ACクライアント要求のための要求ハンドラー

●アンカーを含む、アーチファクトの内容を照会するためのアーチファクト・アクセス機能

●所定のアーチファクト又はアーチファクトのバージョンを呼び出すための、又はリフレッシュするための会話コーディネータの要求に応答し、適切なアンカー・ポイントにナビゲーションし、ACクライアントをオープンするACクライアント・イニシャライザ

【0150】ACIのある特定の実施は、MicrosoftのWord及びJavaモジュールのために特別に作られ埋め込まれたマクロのセットの組合せとして実現され得る。これらのマクロは、文書への新たなアンカーの挿入、古いアンカーの削除、アンカーの移動等を可能にする打鍵及びメニュー選択のセットを通じて、標準のWordの機能性を高める。マクロにより操作される「アンカー」は、専用のActiveX ACアンカー（上述のアンカー参照）である。Javaモジュールは、これらのアンカーからACクライアントへの要求及び情報の流れ、並びにその逆に、ACクライアント及び会話コーディネータから特定のWord文書及びその中のアンカーへ要求を送信する際の要求及び情報の流れを管理する。

【0151】（会話コーディネータ）ACクライアントを実行している各ホスト・コンピュータに常駐する会話コーディネータは、アーチファクトの整合性及びアンカード・カンバセーションと他の会話クライアント（例えば、他のIRCクライアント）との間の通信を維持することに責任がある。また、コーディネータは、会話履歴及びアクセス制御データを含む会話データベースにアクセスする。

【0152】また、会話コーディネータは、限定はされないが、次の属性を含む情報をカプセル化することができる。

アンカード・カンバセーションの一部として共有されているアーチファクトのファイル位置

会話データベース（単数又は複数）の位置（例えば、ファイル・パス、URL（ユニバーサル・リソース・ロ

ケータ)、ソケット識別子)

他の会話クライアントとの通信のための接続情報(例えば、ファイル・パス、URL、ソケット識別子)

【0153】コーディネータの実施は、独立して実行されるプログラム(例えば、Java又はVisual Basicプログラムを含むオペレーティング・システムのウィンドウ)、オペレーティング・システムのエージェント(例えば、Microsoft Agents)、又は所定のアプリケーション内で実施されるコーディネータの機能のために埋め込まれたアプリケーション特有の要素(例えば、Javaのアプリレット、ActiveXコントロール、マクロ定義、スクリプト言語コンポーネント、又は他のこのような埋め込まれたソフトウェア構造)を含む。

【0154】実施の一形態において、コーディネータは以下のコンポーネントを有する。

ACクライアント要求のための要求ハンドラは、イベント・ループであり、このループは要求を待ち、要求されたリソースを提供するか、又はシステムの他の部分(例えば、会話データベース)へこの要求を割り当てる。

接続マネージャは、会話クライアント同士間の会話のフローを開始するイベント・ループである。接続要求が成されると、コーディネータは会話データベースからアクセス制御及びクライアント位置情報を検索し、基礎通信システムを用いて許容される接続を確立する。

【0155】(会話データベース)1つ又は複数の会話データベース(例えば、860)が、会話履歴及びアクセス制御に関する情報を管理する。この情報のための要求/応答処理を提供する任意の分散データベース・メカニズム(例えば、共有ファイル管理、リレーショナル・データベース管理システム)が、アンカード・カンパセーションをサポートするために使用されることが可能である。

【0156】会話データベースは、限定はされないが、以下の属性を含む情報をカプセル化することができる。

会話要素(テキスト、グラフィック、及び他の関連媒体)、コマンド呼び出し、アンカーの変更、アンカーのコンテキスト(例えば、近接テキスト、グラフィック、又はアーチファクトからの位置インジケータ)、会話状況の変化、及び共有アーチファクトのバージョン・データ(例えば、最も最近のバージョン)を含む会話履歴

アクセス制御リスト、ユーザ・アカウント及び特権を含むアクセス制御データ

【0157】実施の一形態において、データベースは以下のコンポーネントを有する。

コーディネータ/ACクライアント要求のための要求/応答ハンドラ

データベース管理のためのユーザ・インターフェース
会話アクセス制御及び履歴要求のためのロー・レベルのデータベース・アクセス方法

【0158】アンカード・カンパセーションは、任意のアプリケーションに配置されることが可能である。以下では、内容が図形であっても、全ての文書の内容を「テキスト」と呼ぶ。上述のように、アンカード・カンパセーションは、限定はされないが、以下の特徴を含む。

●ユーザは、作業文書に会話をアンカー又は添付することができる。

●ユーザは、会話が特定の位置にアンカーされること、(上述の図4乃至7に示されるように)一時的にディタッチされるか、又は完全にディタッチされることを指定することができ、後者によりその会話は文書との関連を解かれる。

●アプリケーションは、そのアプリケーションに関連付けられる(即ち、そのアプリケーションにアンカーされる)会話へのアクセスを提供する。

●個々のアンカード・カンパセーションは、それらがアンカーされる文書へのアクセスを提供する。

●会話コーディネータは、会話及び関連アプリケーションへのアクセスを提供する。メタデータを介して、ユーザは、それらへの様々なビューを有することができる。

●アンカード・カンパセーションは、文書に関する記録メタデータを提供する。例えば、自動的にログが記録されるアンカーの生成、配置及びナビゲーションの履歴、並びに同期又は非同期の共同作業においてユーザにより成される会話を介する。

●アンカード・カンパセーションは、それらのコンテキスト(即ち、アンカード・カンパセーションが、アーチファクト内のそれらの位置に関する所定のデータ及び/又はそれらの周囲のコンテキストにリンクされていること)、又はそれらが配置される場所「に関して認識しており」、この情報をユーザのためにコンテキストを自動的に再生するために使用する。

●アンカード・カンパセーションは、ユーザのための共有の、コンテキストが反映されたアクティビティ・ローケールの生成を自動化する。

●アンカード・カンパセーションは、それらの設定と共に移動することが可能である。

●ディタッチされると、アンカード・カンパセーションは、それらの設定通りには移動しないが、最後に属していた設定を記憶しており、アンカーがリセットされると、それらの設定に自動的に戻ることが可能である。

●アンカード・カンパセーションは、共有画面を実施することなしに、共有コンテキストを提供する。

●ユーザは、動的な、社会に関連し、社会的に構成されたランドマークに基づいて文書をナビゲーションすることができる。

会話がアンカーされる文書は、再表現されることを必要としない。

ユーザは、WYSIWIS(What You See Is What I See)画面に移動することなく、バックグラウンドのビューを

共有しながら、アンカード・カンパセーション空間内で会話及びアクションを共有することができる。

ユーザは、文書自体のコンテキスト内でその文書に関する彼らの会話の履歴にアクセスすることができる。

アンカード・カンパセーションは、共同作業者が同場所に位置しなくても、文書内への同期及び非同期の共同作業による注記事項の作成をサポートする。

アンカード・カンパセーションは、重要なポイントで文書に印をつけるためのメカニズムを提供する。

アンカード・カンパセーションは、(1) 会話への招待を受諾する、(2) 文書を開き、そこに配置されている会話を選択する、又は (3) 会話コーディネータを介して会話にアクセスする、ことにより呼び出されることが可能である。

会話は、関連アプリケーションへのアクセス、及び適切な場合には、そのアプリケーション内の特定のコンテキストへのアクセスを提供する。

アンカード・カンパセーションは、軽い通信ツールのための新たな比喻を具体化する。本発明者らは通信と既存のアーチファクトとの関係を逆にすることから、これを「CSCWの裏返し」と呼ぶ。アンカード・カンパセーションは、会話を作業テキスト内に持ちこむ。

アンカード・カンパセーションは、文書の会話、並びにアプリケーション又は文書レベルのナビゲーションを介して、テキスト・ベースの、作業に関するナビゲーションをフォアグラウンドで実行する。

●アンカード・カンパセーションは、細分性の微細なレベルでの作業テキストの共有を可能にする。アンカード・カンパセーションは、ユーザが単に文書自体ではなく、文書の内容を共有することを可能にする。

【0159】アンカード・カンパセーションを用いることにより、テキスト（又は、例えばグラフィック等の任意の内容）は会話のための位置又は場所を提供する。これは、場所に基づく通信ツールの標準のビューを変える。

【0160】図11は、本発明のアンカード・カンパセーション・システムの実施の一形態の主要コンポーネント間の流れ図を示す。Word文書800は、ActiveXコントロール900にリンクされているアンカード・カンパセーション830を有する。アンカード・カンパセーション830の起動（例えば、関連するサムネイルをダブル・クリックする）時に、コントロール900は1つ又は複数のWordマクロ910（又は、例えば別のActiveXコントロール）を開始する。マクロ910は、会話コーディネータ850に情報（例えば、TCP要求及びUDPメッセージ）を送信する。

【0161】実施の一形態において、会話コーディネータ850は、データベースの関連部分をキャッシュするデータ記憶装置を含む。関連部分は、会話全体、アンカー位置、又は会話コーディネータ850により調整され

る現在の会話に関する他のデータの何れか1つを含む。

【0162】会話コーディネータ850は、もしあればアンカー830により開始されたアンカード・カンパセーションと関連付けられた先の会話情報を検索するため、及び任意の新たな現在の又は同時発生会話情報を格納するためにデータベース・サーバ940（及びメインのリモート・データベース930）と通信する。会話情報は、会話自体と、その会話に関する任意のメタデータ（例えば、アンカー位置、及び／又は他のコンテキスト情報）を含む。また、データベース・サーバは、他の関連及び非関連通信クライアント（例えば、960及び970）と通信する。

【0163】また会話コーディネータ850は、ネットワーク又は、例えばIRCネットワーク950として図示される他の通信媒体／装置と通信し、アンカード・カンパセーションの実際の会話の流れ細目（例えば、会話の招待又は終了）を開始し、維持する。会話の内容を送信するのに十分な、任意の種類の会話クライアント又は媒体が、これらの通信を維持するために利用されることが可能である。

【0164】IRCネットワーク950は、アンカード・カンパセーションに参加する別の位置（ユーザ）（例えば、960）と関連して実行される第二の会話コーディネータと通信する。例として示されるプログラム・フローの他の変形又は変更は、既に当業者には明らかであろう。

【0165】（実施例）以下は、本発明の実施例である。この例では、アンカード・カンパセーション・システムの構成概念を、特にMicrosoftのWordで実施する1つの可能な方法を示すことが意図される。しかしながら、この例はMicrosoftのWord又は本明細書に示される何れの特定の教義にも制限されない。なぜなら、これらの同教義は、幾つか挙げると、ワープロ、スプレッドシート、及びグラフィック・パッケージ等の他のアプリケーションに適用されることが可能であり、本明細書で説明される処理を実行するための如何なる特定の実施の詳細にも依存する必要がないためである。

【0166】アンカード・カンパセーション・システムの実施の一形態は、4つの主要部（及び括弧内に示される実施例の言語）から構成される。

1. ActiveXコントロール・アンカー（VB6）
2. Wordマクロのセット（VBA）
3. ローカルの会話コーディネータ（Java）
4. 単一のメイン・リモート・データベース（Java）

【0167】これらの主要部のそれぞれは、様々な方法（例えば、図11参照）を使用して通信する。アンカー830と関連付けられたActiveXコントロール900は、アンカー830の画面上での状態変化が発生すると、OLEイベント・メカニズムを使用して、イベントを起動する。これらのイベントは、特定のWordマクロを

呼び出し、イベントの機能が必要とする場合には、通信情報が、ローカル・ポートに発信される同等のUDPメッセージに変換される。

【0168】会話コーディネータ850は、必要に応じてローカルのチャット・ウィンドウを更新することによりこれらのメッセージを（例えば、UDPソケット・サーバ928を介して）受信し、これらに応答する。実施の一形態において、会話コーディネータ850はまた、データベース930のレプリカ935を保持する。レプリカ935が変更されると、各変更がHTTP要求（又は他の通信メカニズム）を介してメインのリモート・データベース930に送られる。このデータベースは、ハブとして機能し、要求を他の全ての現在活動状態のレプリカ又はバックアップに送信する。このような更新が基本となっているWord文書の状態を変更することを要求する場合（例えば、異なる装置上のユーザがアンカーを移動又は削除する場合）、Wordのマクロ910は、このような要求を（例えば、それら自体のTCPサーバ920を介して）受信し、それに基づいて動作する。ローカル・マネージャ（会話コーディネータのローカル・インスタンス）は、Wordマクロを使用し、文書の状態（例えば、その文書のファイル名、ID等）を照会する。

【0169】例えば、チャット、即ち会話クライアント820は、IRCプロトコルによりサポートされる。インターネット・リレー・チャット（IRC）プロトコルは、様々な専用のチャット・クライアント（例えば、Windowsプラットフォーム上のmIRC）により幅広く使用されている。各クライアントは、単一のIRCサーバに接続され、このIRCサーバ自体は他のIRCサーバに接続される。メッセージは、クライアントからそれに接続されたサーバへ、そしてIRCネットワーク上の他の全てのサーバに流れ、適切な場合にはそこから各サーバのクライアントに流れる。アンカード・カンパセーション・システム内の各会話クライアントは、関連するIRCチャット・チャンネルにマッピングされ、会話コーディネータは、IRCクライアントとして、並びに会話クライアントから（IRCネットワーク950内の）IRCチャット・チャンネルへ、そしてその逆へ流れるメッセージの導水路として作用する。或いは、会話クライアント（例えば、820）は、IRCネットワーク950に直接通じるそれら自体の個々の接続（例えば、936）を有してもよい。

【0170】図10及び11に示される構造は、会話クライアント（チャット、又はチャット・ウィンドウ）がプラットフォームの範囲全体でサポートされることを可能にする。チャット・ウィンドウの管理及びチャットの内容（現在はJavaアプリケーション内で実施されている）は、チャットがアタッチされるオペレーティング・システム特有のアプリケーションから分離している。これは、アプリケーション会話インターフェース（AC

I）APIをサポートする任意のアプリケーションが、会話コーディネータにより使用されることを可能にする。会話コーディネータからのサポートは、マネージャが異なるアプリケーションをサポートするために統合することができる、共通の追加パッケージを構成するように、独立したパッケージに分割されてもよい。

【0171】（データベースの構造）「データベース」（例えば、メインのリモート・データベース930）は、システムに認識されているエンティティ（構成要素）、アーチファクト、又は作業アーチファクトである全てのチャット、アンカー及び文書の現在の状態を格納する。その実行時の実施は、各エンティティのためのJavaのクラスのインスタンスのセットである。各エンティティは、それらの状態をテキスト形式へパッキング、及びテキスト形式からアンパッキングするためのメソッドのセットによりサポートされる。この形式は、会話コーディネータ及びメインのリモート・データベース930の間でインスタンスを送信するために使用される。同じ形式が、インスタンスをリモートのメイン・データベースのディスク上に又はバックアップ媒体に格納するために使用されることが可能である。

【0172】3つの主なデータベース・クラスは、StickyChat.Document、StickyChat.Chat、StickyChat.Anchorである。文書、チャット又はアンカーの各インスタンスは、対応するユニークなID（例えば、ランダムに生成される）を有し、このIDはシステムにより最初にこのエンティティが生成されたか、又は認識された時に割当られる。これらのクラスにより保持される最も重要な情報は、以下の通りである。各文書に対して、

- 文書ID（文書をユニークに識別する）
- 文書の現在のマスター・バージョンが確認され得る場所のIPアドレス
- IPアドレスにより示されるホスト・コンピュータでの現在のマスター・バージョンのファイル名
- この文書に含まれる全てのアンカーを指し示すアンカーIDのリスト各チャットに対して、
- チャットID（チャットをユニークに識別する）
- このチャットが関連する全てのアンカーを指し示すアンカーIDのリスト（通常は、1つのみ）
- このチャットを使用することを許された人々であるメンバーのリスト
- チャットのタイトル
- チャットが現在アタッチされているアンカーのID（複数のアンカーを有する場合に重要である）。またこれは、活動状態のアンカーIDとして認識される。各アンカーに対して、
- アンカーID（アンカーをユニークに識別する）
- アンカーが配置されることが可能な現在のマスター文書内のあるテキストであるコンテキスト
- このアンカーがリンクされているチャットのチャット

I D

このアンカーが含まれている文書の文書 I D

【0173】各クラスの全ての公知のインスタンスは、3つのテーブル（表）のクラス、StickyChat.DocumentTable、StickyChat.ChatTable、StickyChat.AnchorTableを使用して、各データベースで索引づけられる。このように、アンカー、チャット又は文書に I D が与えられていることにより、このエンティティに関する現在の情報は、関連するテーブル内のインスタンスを参照することにより取得されることが可能である。例えば、アンカー I D によりリンクされた他のテーブルを探索することにより、情報のセット全体を取得することが可能である。

【0174】実施の一形態において、メインのリモート・データベースは、各会話コーディネータの位置で完全に複製される。各会話コーディネータが開始されると、これは HTTP 要求を介してデータベース全体のコピーを要求する。もし、何れのデータベースとも連絡を取ることができない場合は、マネージャはその始動を中止するか、データベースを生成するか、データベース接続無しに機能するか、又は別のフォール・バック・ルーチンを実行し得る。別の実施の形態においては、データベースのより制限されたキャッシングがサポートされ、最も検索されそうなデータ（例えば、アンカー位置）のみを保持する。別の実施の形態では、メインのリモート・データベース 930 のみが保持される。

【0175】データベースのレプリカを起動する各会話コーディネータは、データベースの変更及び更新に応じて HTTP サーバが呼び出す幾つかの標準方法を提供する。「mergeX」メソッド（X は、単数又は複数のアンカー、単数又は複数の文書、単数又は複数のチャットの内の 1 つである）は、データベースがアンカー、文書又はチャットの新たなインスタンスをその現在の状態に追加するか又は組み合わせる必要がある場合に呼び出される。各マージ・メソッドは、更新のソースを示すパラメータを受諾する。マージ・メソッドは通常、（そのソース自体が更新されないように）他のレプリカにこの更新を通知するための「broadcastX」メソッドを呼び出すので、このソースは必要である。

【0176】例えば、新たなアンカーは、ローカルの会話コーディネータにより（例えば、受信された UDP メッセージを介して）ローカル文書内で検出される。新たなアンカー・インスタンスが生成され、ローカルの会話コーディネータの「mergeAnchor」メソッドが、ソースとしてそれ自体の I P アドレス（又は「null」或いはソースを示す他の識別子）と共に呼び出される。会話コーディネータ・データベースは更新され、これは「broadcastAnchor」メソッドを呼び出す。このメソッドは、アンカーを送信可能な形式にパッケージ化し、これを HTTP 要求を介してリモート・データベース・サーバ 940 に送信する。リモート・データベース・サーバの HT

TP サーバは、この要求からアンカーをアンパックし、そのサーバ自体の「mergeAnchor」メソッドをローカルの会話コーディネータの I P アドレスと共に呼び出し、そのサーバ自体のデータベースを更新する。マージ・メソッドは、再度「broadcastAnchor」を呼び出し、このメソッドは、各コーディネータがこの更新のオリジナル・ソースでないことを確認して、リモート・データベースに先にログインしている全てのローカル会話コーディネータへ同じ変更を発信する。

【0177】この議論の多くの部分では、HTTP プロトコル及び UDP メッセージングが、本発明により実施されるべき選択された通信プロトコルとして識別される。しかしながら、当業者により理解されるように、これらの通信には他のプロトコル（可能な例を少しだけ挙げると、X.25、X.400、基本的なハンド・シェイク方式、電子メール等）が利用されることが可能である。

【0178】（アンカー）アンカーは、画面上でのアンカーの状態（アンカーの X、Y 座標及びアンカーが実際に可視であるか否か）、ユニークなアンカー I D、及びアンカーに関連付けられたチャット I D を維持すること、並びにこれらを利用可能にすることに責任がある。アンカーは、この情報に関する変更を 2 つの主な方法で信号を送る。第 1 の方法は、標準の O L E イベントを起動することによるものであり、第 2 の方法は、変更を UDP を介してローカル・ポート（例えば、固定部分（a fixed part））にブロードキャストすることによるものである。当業者には理解されるように、第 2 の方法はより一般的であるが、これらのアンカーの機能性の実施はアプリケーション特有であり異なる。

【0179】本発明者らは、Word 用に ActiveX コントロールとして V B 6（Visual Basic 6）のアンカーを開発した。ActiveX コントロールは、MS プラットフォーム上で ActiveX コントロールが含まれた文書と共に持続され保存される。現在のコントロール（制御）形態は、Microsoft の Word 内のものとして説明されるが、同じ原理は、ActiveX コントロールをサポートする PowerPoint、Excel、バージョン 4.0 以降の Microsoft Internet Explorer により見ることが可能な HTML ウェブ・ページ等の、任意の文書又はアプリケーションに適用される。

【0180】このコントロールは、通常取得しない windows システムのイベントを受信するために、サブクラス化（subclassing）として知られる技法を使用する。詳細には、この技法は、コントロール及びこのコントロールが描き換えられるか又は移動されるたびに生成される WM_PAINT 及び WM_MOVE イベントのコンテナをサブクラス化する。このコントロールは、その図形クリップ領域を照会することにより、このコントロールがもはや可視でないことを検出する。これらのメカニズムには、（a）ユーザが別のウィンドウをアンカーの上に移動して、そのアンカーが隠れてしまう場合、及び（b）メインの wo

rdのウィンドウが画面の周囲にドラッグされる場合には通常、イベントは受信されないのが難点がある。両方の問題を解決するための1つの方法は、アンカーの現在の状態を「ポーリング」するために（アンカーの現在のX、Y座標を取得するため及びそのクリップ領域をチェックするために）タイマーを使用することである。実施の一形態において、ポーリングはActiveXコントロールにより実行されるが、Wordのマクロ又は別のプログラムがこのタスクを実行するように構成されてもよい。

【0181】何れのポーリング方法も性能に関する理想的ではないかもしれないが、ポーリング方法は非常に信頼性が高い。残念ながら、Wordにおいては、Wordにより用いられる中央のタイマーと何らかの衝突／矛盾があるように見える。これは、タイマーがWord文書に関して何れの関数も実行しない場合でも、VB6又はWin32のタイマーにより始動される各イベントごとに、Wordがそのテキスト状態を「リフレッシュ」する（スペル、文法等をチェックする）ことをもたらす。

【0182】(a) に対する1つの解決策は、ユーザがもはやWordを使用しなくなった場合に、これらのアンカーに関連付けられたチャット・ウィンドウの動作の仕方を変更することである。従って、いつアンカーが可視であるか否かを確認するよりも寧ろ、ローカルのマネージャは、そのアプリケーションがもはやフォーカスされていない場合、アンカーを介して特定の文書と関連付けられた全てのチャット・ウィンドウが「最上部」のウィンドウではないことを確認する。Wordウィンドウ内のフォーカスの喪失及び獲得を検出するこの機能は、メインのWordウィンドウ自体をサブクラス化する（アンカード・カンパセーションの実施のためにOLEカスタム・コントロール・ファイルへコンパイルされた別のActiveXコントロール、例えば、「StickyChat.COCX」ファイルへコンパイルされた「StickyMoveTrapper」コントロールを使用することによる）ことを通じて得られる。(b) に対する1つの解決策は、そのウィンドウが動くたびに各アンカー状態を検出し、リフレッシュするためにサブクラス化関数を使用することである。

【0183】ActiveXコントロールは、コントロールが実際にいつ削除されるかを、内部で利用可能なイベントのみを使用して判断することを困難にしている特有のライフ・サイクルをたどる。例えば、コントロールが「無効にされる（破壊される）」前に、「特性の書込み」要求が受信され、これは標準のActiveXコントロール・メカニズムを使用して、その状態を持続することを可能にする。これが再度インスタンス生成される際には、これは同じメカニズムを使用して「特性読出し」要求を受信し、先の状態を取り出すことが可能となる。

【0184】残念ながら、この「書込み」要求の信号が送られている際には、ユーザがこのコントロールを削除している、又はWord文書が閉じられているという理由で

困難が生じる。ライフ・サイクルにおいて、「インスタンス生成」メソッドはアンカーが最初に生成されるときに一度だけ呼び出され、そこでアンカーのユニークなID及びそれに関連付けられたチャットIDが生成される。

【0185】「終了」メソッドは、各ActiveXコントロール上で、それが無効にされる前に呼び出される。更に、コントロールは、もしそのコントロールへの参照カウンタ（即ち、オブジェクトを指す他の変数の数字）がゼロを越える場合には終了しない。これは、ActiveXコントロールがイベント・リスナー（以下を参照）により包まれている場合に、イベント・ラッパーは常にActiveXコントロールを照会するので、終了メソッドが呼び出されないという循環性の問題を生じさせる。

【0186】（Wordマクロ）Wordのマクロは、文書内のアンカーを操作する（追加する、隠す／表示させる等）ためのGUIインターフェースをユーザに提供し、またTCPサーバに基づくインターフェース（繰返しになるが、他のタイプの通信又はインターフェースも利用可能である）を、文書、チャット及びアンカー状態への変更が、開かれている文書及び会話コーディネータによる関連付けられた会話クライアントの管理に反映されることを可能にするために提供する。これらの変更は、新たな位置へのアンカーの移動、所定のテキスト・コンテキスト（アーチファクト内の位置）でのアンカーの作成等を含む。更に、マクロは、文書内の全てのActiveXアンカー・コントロールを「包み」、それらの現在の状態に関するイベントを受信する。最後に、上述のようにマクロは、別のStickyChat（アンカード・カンパセーション）コントロール（実施の一形態においては、StickyMoveTrapperコントロールにより実行される）を用いて、メインのWordウィンドウをサブクラス化し、その画面上の位置及びそのフォーカスにおける変更を受信する。

【0187】これらのマクロを含むWord文書が初めて開かれると、幾つかの「自動」イベントが起動される（文書を開くための「auto_open」及びアプリケーション自体の開始のための「auto_exec」）。マクロは、これらのイベントにトラップを付加し、移動イベント及びフォーカス・イベントを受信するためにWordのウィンドウをサブクラス化する。マクロは、文書内の全てのInlineShape（インライン形状）のワード・オブジェクトを、それらがアンカーであるか否かを確認するためにチェックする。InlineShapeは、WordがActiveXコントロールを包含するために用いるOLEコンテナである。実施の一形態において、アンカー・コントロールは、「StickyChat.Sticky」のOLEクラスを有するInlineShapeオブジェクトとして識別される。確認される各制御ごとに、マクロはVisual Basic (VB) ラッパー・オブジェクトを生成し、これはアンカーOLEイベントの受信及びそのイベントの際に動作するルーチン（例えば、マクロ）へ

のそのイベントの転送に対して責任がある。各VBラッパー・オブジェクトは、文書内のInlineShape Wordオブジェクト及びそのInlineShapeに包含されたStickyChatアンカー・オブジェクトへのポインタを有する。各VBラッパー・オブジェクトは、その文書のための単一の集合内に保持される。

【0188】Wordウィンドウのサブクラス化及び各アンカーの包含に加えて、マクロは第1及び第2の隠れ形態（表示される視覚的表現を持たないコード・オブジェクト）を生成する。この隠れ形態及びそれらの制御は、第1の隠れ形態により生成されるUDPソケット926及び第2の隠れ形態により生成されるTCPソケット・サーバ920等のオブジェクトを（Visual Basicで）実行する1つの方法である。

【0189】UDPソケット926は、アンカー状態への変更を発信する。第1のTCPコントロール（TCP

Word マクロによりサービスされる TCP 要求

INFODOONLY	文書に関する情報（ファイル名）を取得する。
SHOW	特定文書内の特定のアンカーが画面上に表示されることを確実にする。
INFOALL	この Word インスタンス内で開かれた全ての文書及びそれぞれに含まれたアンカーに関する情報を取得する。
INFOANC	特定の文書内の特定のアンカーに関する情報（X、Y座標、画面上の可視性、チャットID及び内容）を取得する。
INFODOC	全てのアンカー情報を含む、文書に関する情報を取得する。
ADDANCHOR	特定のコンテキスト（テキスト・ストリング）に所定のID及びチャットIDと共にアンカーを追加する。
MOVEANCHOR	既存のアンカーを文書内の新たなコンテキスト（テキスト・ストリング）に移動する。
DELETEANCHOR	文書からアンカーを削除する。
CHANGEAID	文書内のアンカーのアンカーIDを変更する。
CLOSEDOC	文書を閉じる。
SAVEDOC	文書を保存する。
SPLITDOC	ローカル・マネージャがこのIP上のマスター文書のファイル名とこの文書IDとの不一致を検出する際に呼び出される。このような状況において、ユーザは、この文書が新たなマスター文書であるか、又はコピーであるかを指定することができる。この文書がコピーである場合は、この呼出しが成され、全てのアンカーを除去し、新たな文書IDを割当てて、
HASDOC	文書が所定のファイル名と共に開かれ、保存を必要とする場合に、文書情報を戻す。
REFRESHANCHORS	各アンカーに対し、その現在の画面上の位置及び状態をリフレッシュさせる。
ENABLEANCHOR	アンカーを対話可能状態又は不可能状態にする（不可能状態はアンカーをグレーにし、クリックに対して応答しない）。

ソケット・サーバ920）はサーバであり、ローカル・マネージャ（この例では会話コーディネータ850）から入ってくる全ての要求を受信する。

【0190】アンカーを含む文書が開かれると、Wordマクロは、そのTCPサーバ上の外部変更及び要求を待ち受け（関連するイベントを受信するために待機し）、UDPを介してアンカーの変更をリモートのローカル・ポートに送信し、ユーザが新たなアンカーを追加すること、又は文書内の既存のアンカーを変更することを可能にするためのメカニズム（例えば、メニューによる選択又はボタン）を提供する。

【0191】実施の一形態のために、TCPサーバへの外部要求が表2に示される。全ての要求は、必要な文書ID及び／又は要求されるアンカーIDを渡す。

【表2】

【0192】アンカー及び文書状態への変更に応じて、 *メッセージが表3に示される。
マクロによりローカル・マネージャに発信されるUDP * 【表3】

Word マクロにより送信されるUDPメッセージ

Removed	アンカーが文書内にもはや存在しない。
Click	アンカーがクリックされた。
XY	アンカーのXY位置が変更された。
Vis	アンカーの可視性が変更された。
Open	所定のIDを有する文書が開かれた。
Close	所定のIDを有する文書が閉じられた。
Activated	文書が活動状態にされた（フォーカスを有する）。
Deactivated	文書がフォーカスを失った。

【0193】マクロが、先に包含されたアンカーが無効なInlineShapeを有する（即ち、ユーザにより削除された）ことを検出するたびに、削除されたことを伝えるUDPメッセージが生成される。マクロは、「書込み」メソッドが、削除又は保存される前にアンカーに呼び出される（これがイベントの起動をもたらす）という事実を利用する。このイベントが発生すると、Wordのアプリケーション・タイマー（全てのマクロが共有するのは1つ 20 だけである）が、数秒後に始動するように設定される。タイマーが始動すると、全てのアンカーが有効性を確認され、有効なInlineShapeコンテナを持たないすべてのアンカーは、削除されたと考えられる。ユーザがアンカーを文書の周囲に「ドラッグ」すると、アンカー及びそのアンカーのInlineShapeの両方が無効にされ、そのドラッグ操作が終了すると、（新たなInlineShapeと共に）再生されることは、特筆に価する。このオペレーションは、MicrosoftのWordアプリケーション自体により実行される。

【0194】マクロが、サブクラス化されたWordのウィンドウを移動するイベント（例えば、上述のStickyMoveTrapperコントロール）を傍受するたびに、各アンカーはアンカーの画面上の状態をリフレッシュすることを強いられ、これは次に、「XY」メッセージで送られるXY変更イベント等を生成することをもたらす。

【0195】メインのWordウィンドウの活動化及び非活動化もまた、メインのローカル・マネージャが、Word文書内のアンカーと関連付けられたチャット・ウィンドウの「最上部」フラグを変更することを可能にするように、トラップが付加され、UDPメッセージを生成する。例えば、ユーザが文書に取り組んでいるか、その中味をスクロールしているか、又はウィンドウをドラッグしている間は、チャット・ウィンドウは、その文書上のアンカーの位置に現われるように再配置されることが可能である。ユーザが別のアプリケーションに切替えると、チャット・ウィンドウの最上部フラグは、チャット・ウィンドウが全てのものの上に表示されるのを防ぐために不能状態にされる。

【0196】（ローカル会話コーディネータ）ローカル 50

会話コーディネータは、各ユーザ装置上で実行され、アンカーの座標に配置されるチャット・ウィンドウ（例えば、820）の生成及び管理に責任がある。ローカル会話コーディネータは、実施の一形態では、メインのデータベースのレプリカを保持し、少なくとも1つの会話クライアントとIRCネットワーク950との間のIRCメッセージを調整し、他のユーザからの会話要求に応える。

【0197】会話コーディネータは、システムの異なる面で作用する4つのメインのスレッドを実行する。UDPスレッドは、ローカル文書及びアンカーから入ってくるUDPメッセージを待ち受ける。TCPサーバ・スレッドは、IRCメッセージを送受信するためのIRCクライアント・インターフェースとして使用される。もう1つのTCPサーバ・スレッドは、単純なHTTPサーバのように動作するように使用され、このサーバは、メインのリモート・データベースと通信するため、及び他のユーザから（それらの会話コーディネータを介して）文書要求を受諾するために使用される。最後に、一次開始スレッド（primary starting thread）はGUIを実行し、ウィンドウの更新及び生成等を行う。或いは、スレッド間での呼び出しメソッドが利用されてもよい。

【0198】マネージャが、そのローカル文書（ローカルのファイル記憶装置上の文書）の1つに対する要求を受信すると、このマネージャは、その文書が存在するか、及びその文書が現在開かれているかを確認するためにチェックする。もしこの文書が（Wordで）開かれており、保存されることが必要である場合、マネージャはユーザに、その文書の現在の最新版が要求されている旨を通知し、ユーザが現在開かれているバージョンを保存したいか否かを尋ねる。次にマネージャは、このファイルを要求した他のローカル・マネージャにこのファイルを戻す。文書取得要求全体は、正式なHTTPサーバが文書を戻すのとはほぼ同じ方法で応じられる。実際に、もしこの要求が通常のブラウザからのURL要求のように擬態されると、このブラウザは簡単にファイルを取得し、ユーザはこれを開くことができる。

【0199】実施の一形態において、会話コーディネー

タ及びデータベース・サーバ（以下参照）の両方により使用される I R C クライアント・パッケージは、Java I R C パッケージの「リレー」である。これは、I R C プロトコルを使用してコマンド及びメッセージを送受信する方法を理解するクラスのセット（Mozillaの認可の下でリリースされている）である。このクラスを使用するために、会話コーディネータは I R C アプリケーション・インターフェースを実施するクラスを提供する。このクラスは、ユーザが加わり得る特定のチャンネルよりも寧ろ、一般の I R C サーバへの接続に関するイベントを受信する。リレー・パッケージが特定チャンネルのためのメッセージを受信すると、それは「IRCView」インターフェースを実行する別のクラスのインスタンスのための I R C アプリケーション・クラスを求める。チャンネルごとに 1 つの IRCView クラス・インスタンスが存在する。次に、リレー・パッケージは、ユーザがメンバーである間は、このチャンネルに関連する I R C アプリケーションにより戻されるそのビューに関する様々なコールバック（人が加わる、離れる、メッセージが送信されている等）を呼び出す。

【0200】実施の一形態において、会話コーディネータ内の I R C クライアント及びデータベース・サーバにより用いられる I R C サーバは、リモートの公用サーバである。ユーザ名とチャンネル名の衝突／矛盾を避けるために、I R C のビュー及びアプリケーション・クラスは、I R C チャット・システムと対話する際に、共通の文字順序を、マネージャにより使用される全てのチャンネル並びにユーザに付加（prepend）する。実施の一形態において、I R C チャンネルは排他的である（パスワードにより保護されるか、又は招待されたユーザのみ進入を許可することによる）か、或いは、排他性は強要されなくてもない。

【0201】各ローカル会話コーディネータは、例えば、ローカルのデータ記憶装置 935 に記憶された、ユーザ装置上で現在開かれている Word 文書及びアンカーに関するローカル情報、並びにユーザが実際に活動的に参加している（従って、対応するチャット・ウィンドウが開かれている）全てのチャットを保持する。これらのリストは、ローカル・データベースのコピーを更新するだけでなく、様々な Word ルーチン及びチャット・ウィンドウに、それらに関わるデータベースの特定部分への変更を通知するように、mergeX関数により使用される。

【0202】例えば、アンカーが別のローカル・マネージャから文書に追加又は移動されると、HTTPサーバは、mergeAnchor関数を呼び出す。マージ・ルーチンは、このアンカーが現在開かれている Word 文書と一致する文書 ID に属しているかを確認するためにチェックする。もし、その Word 文書と一致する文書 ID に属しているならば、Word 文書は更新される必要がある（アンカーが追加されるか、又は移動される）。ローカルで開かれ

ているチャットに対しても、同様のチェックが成される（メンバーが追加されている等）。

【0203】実施の一形態において、文書のコピーとマスター・バージョンとの間の同期エラーは、コピーの会話コーディネータがその文書内の特定のコンテキストにアンカーを挿入又は移動しようとするたびに検出される。もしこのコンテキストが見つからない場合は、明らかにこのコピーはマスター・バージョンよりも古いバージョンであり、ユーザは、その文書の最新版を取り出したいかを尋ねられる。会話コーディネータは、ある文書がマスターであることを知っているが、文書とデータベースとの全ての矛盾を同期の問題として扱う。これは、もしユーザがアンカーをマスターのコピー内のある新たなテキストに追加する場合、「マスター」がこの「マスター」のコピーと同期しなくなることをもたらし得る。このような状況では、マスターの実際の現在の内容が正しいと仮定され、データベース内の状態を上書きする。

【0204】ユーザ名、電子メール・アドレス及びフルネーム等のユーザ・プリファレンス（preference）が、システムにより使用される他の可変パラメータ（I R C 接頭部（prefix）、サーバ及びポート、データベース・サーバの I P 及びポート等）と共に普通テキスト・ファイルに格納される。このファイルの名前は、コマンド・ラインで会話コーディネータに渡されることが可能である。或いは、何れのファイルも指定されない場合は、現在の作業ディレクトリ内の「prefs.txt」のデフォルト名が渡されてもよく、又は他のプリファレンス・メカニズムが使用されてもよい。

【0205】（リモート・データベース・サーバ）リモート・データベース・サーバ 940 は、全ての保存された文書、アンカー、チャット情報を含み、「正確な」バージョンと考えられるメインのリモート・データベース 930 を保持する。即ち、リモート・データベース・サーバ 940 は、誰が（データベースの動作中のレプリカに）ログオンしているかを認識し、後の探索及びブラウジングのために、全てのチャットの会話をモニターし、ログを記録する。

【0206】チャットの会話及びアクションのログの取得は、特殊なデータベース・ユーザとして I R C に接続することにより行われる。チャットが生成されるか、又はローカル・マネージャを介して「一般の」ユーザがチャットに参加するたびに、特殊なデータベース・ユーザも招待される。招待状は、I R C クライアント・インターフェースを介してデータベースに届き、このデータベースがこれを受諾し、チャンネルに加わる。各チャンネルごとに、データベースは G U I の無い、単純な I R C ビューを生成し、これは単にそのチャンネル内の全てのアクティビティをローカルのログ・ファイルにログを記録する。チャンネル内にデータベース・ユーザ以外にもユーザが存在している限り、リモート・データベースは

10

20

30

40

50

そこに留まる。データベース・ユーザが、そのチャンネル内の最後のユーザになると、データベース・ユーザは自動的に退去する。

【0207】一般のデータベースに関するHTTP要求(mergeX等)に加え、リモート・データベース・サーバはまた、システムのユーザに関する要求(誰がどこにログオンしているか、誰がチャットに招待され得るか等)を処理する。

【0208】クラスの多くがローカル会話コーディネータ(例えば、850)とリモート・データベース・サーバ940により共有されているので、同一のJavaソース(及びWindowsの実行形式ファイル)が両方で使用されることが可能である。第1のコマンド・ライン・パラメータは、それらを区別するために用いられ、その.exe(実行形式ファイル)がいつ最初に実行されるかが指示される。もし、パラメータが何もないか、又は第1のパラメータが「マネージャ」であれば、その実行形式ファイルが会話コーディネータとして開始される。もし、第1のパラメータが「サーバ」であれば、それはメインのデータベース(例えば、リモート・データベース・サーバ940)として開始される。もしあれば、第2のパラメータは、その会話コーディネータにより使用されるブリファレンス・ファイル(上記参照)へのパスである。

【0209】(全ての統括:チャットの貼り付け方法)Elizabethは、彼女の装置でローカル・チャット・マネージャを起動する。マネージャは、メインのリモート・データベースにログインし、現在のデータベース状態のコピーを受信する。またこのマネージャは、チャット・チャンネルが生成されるであろうIRCサーバと接続する。

【0210】Elizabethは、StickyChatマクロのセットを含む新たなWord文書を開く。これらのマクロは、付加的なツールバーを提示し、彼女が新たなアンカー及びチャットをこの文書に挿入することを可能にする。Elizabethはあるテキストを追加し、そのテキストの挿入ポイントでツールバーを用いてアンカーを挿入する。マクロは、その文書のIDを確認するためにチェックするが、その文書がまだIDを持っていないので、新たなIDが生成され、その文書のワード変数(これらは、保存と保存の間で持続性である)に加えられる。このIDは、持続性であり、Wordによりその文書と共に格納され、その後その文書を開く全てのWordアプリケーションにとって常に利用可能である。マクロはまた、この文書がディスクに保存されるかを確認するためにチェックし、アンカーを追加する前にElizabethに保存するように促す。マスター文書は、文書ID、そのファイル名及びIPアドレスの組合せにより識別されるので、保存は必要である。次に、マクロは、StickyChatアンカーをその中に有する新たなInlineShapeを生成し、この形状を、内部マクロ集合に追加されるイベント受信オブジェクトにより

包む。次にアンカーは、そのアンカー自体をインスタンス生成し、そのコンテナ(そのInlineShape)を位置及び状態の変化ごとにサブクラス化し、新たなアンカーID及びチャットIDを生成し、多数のイベント(画面位置の変更及び視界の変更等)を生成する。ラッパー・オブジェクトはこれらのイベントを受信し、マクロはこれらのイベントに対応するUDPメッセージを生成する。

【0211】Elizabethのローカル・マネージャは、これらのUDPメッセージ・イベントの第1のイベントを受信し、このIDを有する文書がデータベース内で以前に確認されているかをチェックする。この例では、以前に確認されていないので、マネージャは、この文書に関する詳細(その文書内の全てのアンカーに関する情報も含む)のためにマクロのTCPサーバを介してWord文書を照会し、対応する文書及びアンカー・オブジェクトを生成する。もし、複数のアンカーが存在する場合、複数のアンカー・オブジェクトが生成される。その後、これらの新たなオブジェクトは、メインの文書及びローカル・データベース内のアンカー・テーブルに追加され、この変更はHTTP要求を介してリモート・データベースに送信される。リモート・データベースは、そのテーブルにこれらを追加し、他の現在接続されている全てのマネージャ(このメッセージのソースは含まない)(この例のこの時点では、該当するものなし)に同更新をブロードキャストする。

【0212】オリジナルのUDPメッセージは処理を続けられ、アンカーIDが既に認識されているかを確認するためにチェックする。もし、認識されていない場合は、新たなアンカー・オブジェクトが生成される必要がある。この例では、アンカーIDはデータベースのアンカー・テーブル(上記参照)内に対応するオブジェクトを有するが、このアンカーに関連付けられるチャットIDは対応するオブジェクトを有さない。マネージャは、新たなチャット・インスタンスを生成するために処理を続け、ローカル・データベースのレプリカにそれを追加する(そして、リモート・データベースにこの更新を発信する)。マネージャは、チャットのためのsticky chatウィンドウを生成し、送られてくるチャット・メッセージを受信するための新たなIRCチャンネル・ビューを生成し、チャットIDと同じIDを有するIRCチャンネルに加わる。チャンネルへの参加に加えて、マネージャはIRC招待要求を特殊なリモート・データベースIRCユーザに送信する。このデータベースは、自動的にこの要求を受諾し、チャット・チャンネルのログの記録を開始する。

【0213】マネージャは、オリジナルUDPメッセージの処理を終了する。メッセージが視界の変更(この例では、初めてアンカーが可視状態にされる)であった場合は、マネージャは上述のように生成されたチャット・ウィンドウを表示する。メッセージが位置変更であった

場合は、マネージャはこのチャット・ウィンドウの画面上の位置を更新する。ElizabethがメインのWordウィンドウを周辺にドラッグするか、又はこの文書をスクロールすると、マネージャはアンカーの位置に関する変化を受信する。この時点では、この文書ID内のこのアンカーIDのためのエントリがこのチャットIDと共にデータベースに存在するので、UDPレシーバは、その後の画面上の全ての変更のための集中的な処理を必要としない。

【0214】Elizabethが新たなチャット・ウィンドウに数行タイプ入力すると、各行はマネージャにより、ウィンドウに表示されるチャットと関連付けられたチャンネルに送信される。まだ他に誰もこのチャットに招待されていないので、受信者は、リモート・データベース及びこのユーザ用のチャンネルのために生成されたIRCビューのみである。リレー・パッケージがIRCサーバからメッセージを受信すると、リレー・パッケージはそのメッセージを正しいIRCビュー・インスタンスに転送し、このIRCビュー・インスタンスはそのメッセージをチャットのウィンドウに追加する。實際上、Elizabethのメッセージは、Elizabethがそれをタイプ入力するときに直接チャット・ウィンドウに追加されることなく、寧ろIRCチャンネルを介して送られる。

【0215】第2のユーザ、Joelは、別の装置で作業をしており、彼自身のローカル・マネージャを始動させ、これはIRCにログインし、彼とリモート・データベースをつなぐ。

【0216】Elizabethは、その文書に関して共同作業をするためにJoeを招待したいので、彼女はチャット・ウィンドウのメニューから招待オプションを選択する。マネージャは、HTTP要求を使用して、リモート・データベースから既知のユーザのリストを取り出す。このリストが表示され、ElizabethはJoeを選択する。Joeがそのチャット・インスタンス内のメンバー・リストに追加され、この変更がリモート・データベースに送信され、このリモート・データベースからJoeのマネージャに送信される。Elizabethのローカル装置はまた、IRCプロトコルを使用してJoelに招待要求を送信する。IRCサーバは、この招待をJoeのローカル・マネージャに配信し、このローカル・マネージャはその文書への招待を彼に通知するダイアログをポップアップさせる。

【0217】Joelはこの招待を受諾し、彼のマネージャは、Elizabethの装置上のこのチャットに関連付けられたマスター文書のためのHTTP取得要求を送信する。文書オブジェクトは、そのアンカー・オブジェクトの文書ID特性から取得され、アンカー・オブジェクトは、そのチャットのアンカーID特性を使用して取得される。要求される文書の実際ファイル名及びIPは、文

書のオブジェクト内で利用可能である。Elizabethのマネージャは、送られてくる要求を受信し、別のユーザがその文書の現在の状態を要求しているので、その文書をディスクに保存するようにElizabethに促す。Elizabethは、保存ダイアログに対し「はい」と応え、マネージャはWord文書のTCPサーバにSAVEDOC（文書保存）要求を送信する。このマネージャはJoeのマネージャにファイルを送る。

【0218】Joeのマネージャが、彼のローカル・ファイル記憶装置上の一時フォルダに転送された文書を一旦有すると、マネージャは、そのファイルが開かれる前にそのファイルがどこに置かれるべきかをJoelに尋ねる。Joelがファイルを選択し、一時ファイルは移動される。この文書を開く前に、マネージャは、そのローカルの「goto」テーブルに特別なエントリを追加する。このテーブルは、特定の文書が最初にローカルで確認されたときに、特定のアンカーに移動されるべきことを示す。次にマネージャは、転送されたWord文書を開き、招待受諾応答を終了する。

【0219】この文書が内部のWordマクロと共に開かれると、マクロ／文書は、「open」UDPメッセージを送信し、これはJoeのマネージャにより受信される。マネージャは、この文書及び文書中の全てのアンカーのためのエントリをデータベースから発見することができるが、この文書のための「goto」リスト内に未登録のエントリを有する。マネージャは、特定のアンカーIDのためのSHOW（表示）要求をTCP要求を介してこの文書のローカルWordマクロに送信する。

【0220】Wordのマクロは、このアンカーを画面上で利用可能にすることにより応答し、これは、このアンカーの状態における変化に関する様々なUDPメッセージを始動させる。オリジナルのマネージャと同様に、（チャットはデータベース内には存在するが）マクロはこのチャットID用のローカルなチャット・ウィンドウをまだ生成していないので、ウィンドウを生成し、チャットに対応するチャンネルのためのIRCビューを生成するために処理を続け、それに加わる。IRCはこの参加をこのチャンネル内の全てのメンバー（Elizabeth及びリモート・データベース）に信号で知らせ、2人のユーザはこの時点で会話することができる。

【0221】表4乃至7は、対応するメカニズム（例えば、上述のActiveXコントロール）により開始されるマクロ又は他のプログラム（例えば、Microsoft Visual Basic for Applications内のサブルーチン（サブ（プロシージャ）及び関数））として実施され得る幾つかのアンカー・オペレーションのためのプログラム設計言語（PDL）を例示している。

【表4】

アンカーの追加

```

Subroutine addAnchorAt(doc As Document, aid As String, cid As
String, context As String) As Boolean
    Declare myRange As a Microsoft Word Content Object (i.e.,
Range)
    'Find the area in the document to insert the anchor
    'by looking for matching text (what we call the context)
    Set myRange = findContext(doc, context)
    If myRange Is a valid document location Then
        Call doc.addNewSticky(anchorID, chatID)
    End If
End

```

【表5】

会話クライアント (StickyChat と呼ばれる) の追加

```

'To the MS Word `document' class we have added the following
subroutine
Subroutine addNewSticky(anchorID As String, chatID As String )
    Declare thesticky as a Sticky Chat Object
    Declare theresults As A Microsoft Word InlineShape Object

    'Create A new MS Word InlineShapes
    'With an OLEControl of type "StickyChat.Sticky"
    Set result = InlineShapes.AddOLEControl("StickyChat.Sticky")
    Set thesticky = result.OLEFormat.Object

    thesticky.id = anchorID
    thesticky.ChatID = chatID

    'Generate a Context_Changed event to let the system know there
    ' is a new anchor here
End

```

【表6】

30

アンカーの削除

```

Subroutine deleteAnchor(doc As Document, anchorid As String) As
Boolean
    Declare anc As A Microsoft Word InlineShape Object
    'Look up the anchor by its ID
    Set anc = findAnchor(anchorid, doc)
    If anc is actually a valid Anchor Then
        'remove the anchor from it's Microsoft Word wrapper object
        ActiveDocument.unwrapByID (anchorid)
        ' DIE ANCHOR DIE!
        anc.Delete
    End If
End

```

【表7】

アンカーの移動

```

Subroutine moveAnchor
    'Copy the anchor data as temporary data
    'Delete the anchor from its old location
    'Create a new anchor at the new location
    'Reset the anchor data to the old values in temporary data End
End

```


【0222】上記PDL（表4乃至7）のそれぞれは、最終的にコーディングされると、上述のようにユーザがMicrosoftのWord内のアンカーをクリックした際に、イベントがアンカード・カンバセーションを実行するための作動状態のJavaコンポーネントに渡されるようにリンクされる。通信は、UDP及びTCP通信による。Javaのチャット・ウィンドウが生成され、アンカーは、会話コーディネータと通信してそのアンカーの位置を示し、会話コーディネータがチャット・ウィンドウをそのアンカーの近位置に配置することを可能にする。

【0223】上記実施例、及び更に上述された他の教義のそれぞれを含む本発明は、コンピュータ業者には明白であるように、従来の汎用又は特殊デジタル・コンピュータ、或いは本開示の教義に従ってプログラムされたマイクロ・プロセッサを使用して簡便に実施されることができる。

【0224】適切なソフトウェアのコーディングは、ソフトウェア業者には明らかであるように、本発明の開示の教義に基づいて、熟練のプログラマにより容易に準備されることが可能である。また本発明は、当業者には容易に理解されるように、特定用途向け集積回路を準備することにより、又は従来の構成回路から成る適切なネットワークを相互接続することにより実施されてもよい。

【0225】本発明は、本発明の任意の処理を実行するようにコンピュータをプログラムするために使用されることが可能な、命令が中に格納された記憶媒体（単数又は複数）であるコンピュータ・プログラム・プロダクトを含む。記憶媒体は、限定はされないが、フロッピー（登録商標）・ディスク、光ディスク、DVD、CD-ROM、マイクロドライブ、及び磁気光学ディスクを含む任意のタイプのディスクと、ROM、RAM、EPROM、EEPROM、DRAM、VRAM、フラッシュ

・メモリ装置、磁気又は光カード、ナノ・システム（分子メモリICを含む）、又は命令及び／又はデータを格納するのに適した任意のタイプの媒体又は装置を含むことができる。

【0226】コンピュータが読取り可能な媒体（単数又は複数）の何れかに格納される本発明は、汎用／特殊コンピュータ又はマイクロプロセッサのハードウェアを制御するため、及びそのコンピュータ又はマイクロプロセッサが、本発明の成果を利用して人であるユーザ又は他のメカニズムとの対話を可能にするためのソフトウェアを含む。このようなソフトウェアは、限定はされないが、装置のドライバ、オペレーティング・システム、及びユーザ・アプリケーションを含むことができる。最終的には、このようなコンピュータが読取り可能な媒体は更に、上述のような本発明を実行するためのソフトウェアを含む。

【0227】汎用／特殊コンピュータ又はマイクロプロセッサのプログラミング（ソフトウェア）に含まれるのは、本発明の教義を実施するためのソフトウェア・モジュールであり、限定はされないが、作業アーチファクトへのアンカーの挿入、様々なアプリケーションのアプリケーション・プログラミング・インターフェースとの通信、通信及び通信クライアントの開始、会話又は通信クライアントの作業アーチファクト内の対応アンカーとの相対位置の維持、会話の検索及びログの記録、通信依頼の要求及び処理、接続の管理、アプリケーションの開始及びアーチファクトのダウンロード、並びに本発明の処理の結果の表示、格納、又は通信を含む。

【0228】実施の一形態において、本発明は、Java言語により実施され、表8乃至10に示されるようなクラス階層を有する。

【表8】

10

20

30

クラスの階層

```

class java.lang.Object
    class StickyChat.Anchor
        class StickyChat.manager.AnchorWrapper
    class StickyChat.AnchorTable
    class StickyChat.Chat
    class StickyChat.relay.ChatEngine (implements
java.lang.Runnable, StickyChat.relay.IChatEngine)
    class StickyChat.relay.ChatOptions (implements
java.io.Serializable)
    class StickyChat.ChatTable
    class StickyChat.relay.ChatViewManager (implements
StickyChat.relay.IChatViewManager)
    class StickyChat.util.Codecs
    class StickyChat.Dbg
    class java.util.Dictionary
        class java.util.Hashtable (implements
java.lang.Cloneable, java.io.Serializable)
            class StickyChat.manager.GotoList
            class StickyChat.manager.LocalChatTable
            class java.util.Properties
            class StickyChat.util.Options
                class
StickyChat.server.ServerOptions
            class StickyChat.UserTable
            class StickyChat.manager.WordDocIDList
    class StickyChat.Document
    class StickyChat.DocumentTable
    class StickyChat.net.HttpRequestHandler (implements
java.lang.Runnable)
        class StickyChat.net.HttpBasicHandler
        class StickyChat.manager.HttpManagerHandler
        class StickyChat.server.HttpServerHandler
        class StickyChat.manager.HttpFileHandler
    interface StickyChat.relay.IChatApp
    interface StickyChat.relay.IChatEngine
    interface StickyChat.relay.IChatView
    interface StickyChat.relay.IChatViewManager
    class StickyChat.manager.IRCApp (implements
StickyChat.relay.IChatApp)

```

【表 9】

```

class StickyChat.server.IRCApp (implements
StickyChat.relay.IChatApp)
class StickyChat.manager.IRCView (implements
StickyChat.relay.IChatView)
class StickyChat.server.IRCView (implements
StickyChat.relay.IChatView)
class StickyChat.relay.IdentServer (implements
java.lang.Runnable)
interface StickyChat.manager.InviteAcceptReceiver
interface StickyChat.manager.InviteEventReceiver
class StickyChat.manager.LocalChat
class StickyChat.relay.MainStub
interface StickyChat.net.ManagerInterface
class StickyChat.Member
class StickyChat.MemberTable
class StickyChat.util.Misc
interface StickyChat.manager.ProgressReceiver
class StickyChat.relay.RCTest
class StickyChat.net.RequestDispatcher
class StickyChat.manager.HttpRequestDispatcher
class StickyChat.server.HttpRequestDispatcher
class StickyChat.net.ServerCmd
class StickyChat.net.BasyServerCmd
class StickyChat.relay.ServerInfo (implements
java.io.Serializable)
class StickyChat.util.ShellExec
class StickyChat.net.SocketServer (implements
java.lang.Runnable)
class StickyChat.Start
class StickyChat.Tables
class java.lang.Thread (implements java.lang.Runnable)
class StickyChat.net.udpSocket
class java.lang.Throwable (implements
java.io.Serializable)
class java.lang.Error
class StickyChat.manager.RemoteError
class StickyChat.manager.RemoteAppError
class StickyChat.manager.RemoteNetError
class java.lang.Exception
class StickyChat.util.ShellException
interface StickyChat.net.UdpEventReceiver

```

【表10】

```

class StickyChat.User
class java.util.Vector (implements
java.lang.Cloneable, java.io.Serializable)
class StickyChat.AnchorIDList
class StickyChat.manager.WPCinviteEventReceiver
(implements StickyChat.manager.InviteEventReceiver)
class StickyChat.manager.WFCudpEventReceiver
(implements StickyChat.net.UdpEventReceiver)
class StickyChat.manager.Word
class StickyChat.manager.WordCmd
class StickyChat.manager.WordDoc

```

【0229】明らかに、本発明の多数の変更及び変形が、上述の教義の観点から可能である。従って、本発明は、添付の請求項の範囲内で、本明細書に明確に記載されたものとは異なるように実行されることが可能である

ことが理解されるであろう。

【図面の簡単な説明】

【図1】MUDチャット・ウィンドウを含む幾つかのオープン・アプリケーションを有するコンピュータ・ディ

スプレイである。

【図 2】MASSIVE-1の 3 次元環境を共有するブラックボードと化身とを共に示す図である。

【図 3】本発明のアンカード・カンパセーションを示す図である。

【図 4】(A) は非活動状態のアンカード・カンパセーションを示す図であり、(B) は活動状態のアンカード・カンパセーションを示す図であり、(C) は関連アプリケーション及びアーチファクトから切り離されたアンカード・カンパセーションを示す図である。

【図 5】アンカード・カンパセーションの挿入を示す図である。

【図 6】アンカード・カンパセーションの挿入に関連する招待を示す図である。

【図 7】挿入されたアンカード・カンパセーションを示す図である。

【図 8】それぞれがアンカード・カンパセーションの存在を指し示す複数のアンカーを有する文書（アーチファ

クト）を示す図である。

【図 9】視覚キューが描写された文書（アーチファクト）の概略表示である。

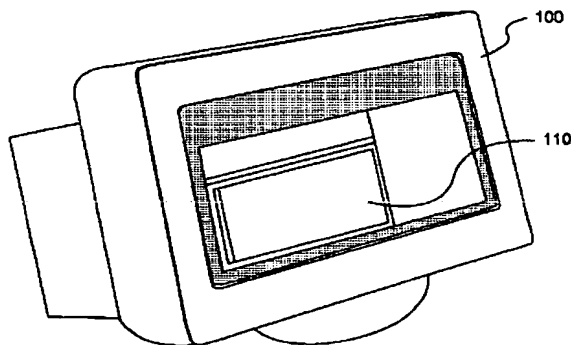
【図 1 0】本発明の実施の一形態を実施するための主要コンポーネントのブロック図である。

【図 1 1】本発明の実施の一形態を実施するための主要コンポーネント及び通信リンクのフロー・ダイアグラムである。

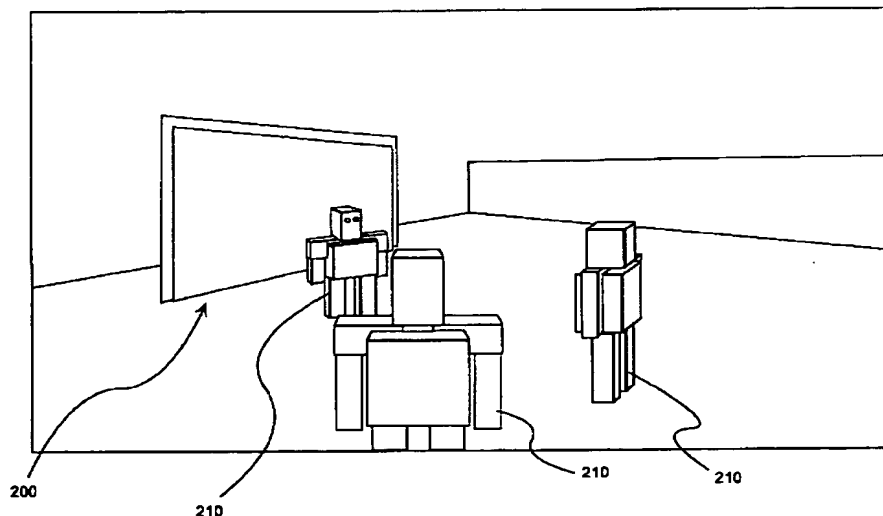
【符号の説明】

8 0 0	作業アーチファクト
8 1 0	アプリケーション
8 2 0	アンカード・カンパセーション・クライアント
8 3 0	アンカー
8 4 0	アプリケーション－会話インターフェース
8 5 0	会話コーディネータ
8 6 0	会話データベース

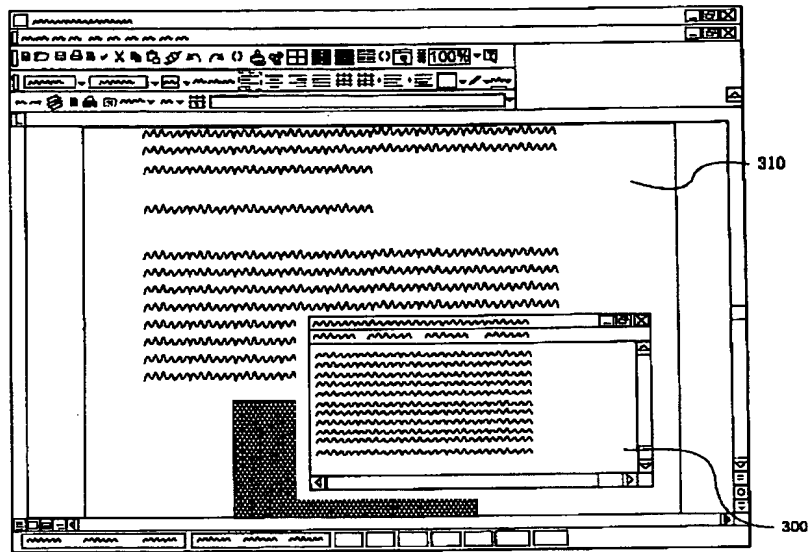
【図 1】



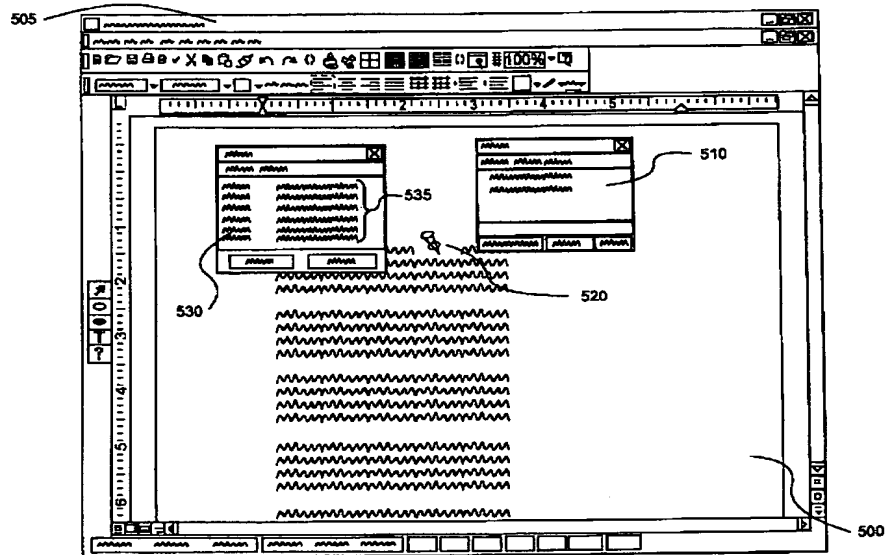
【図 2】



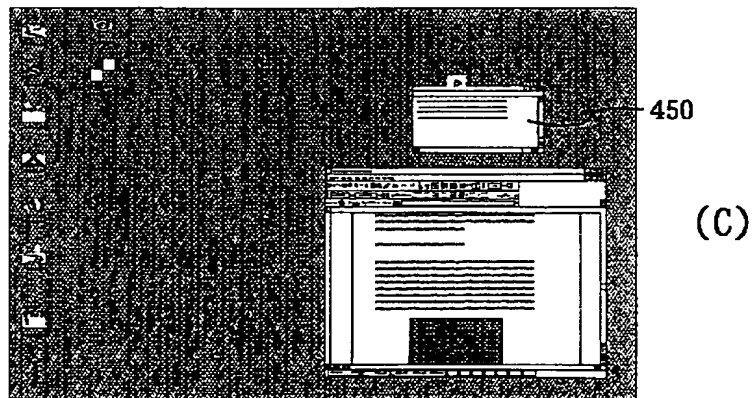
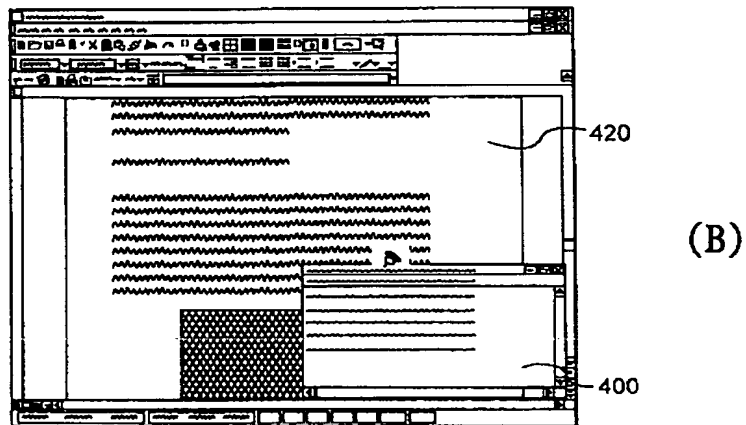
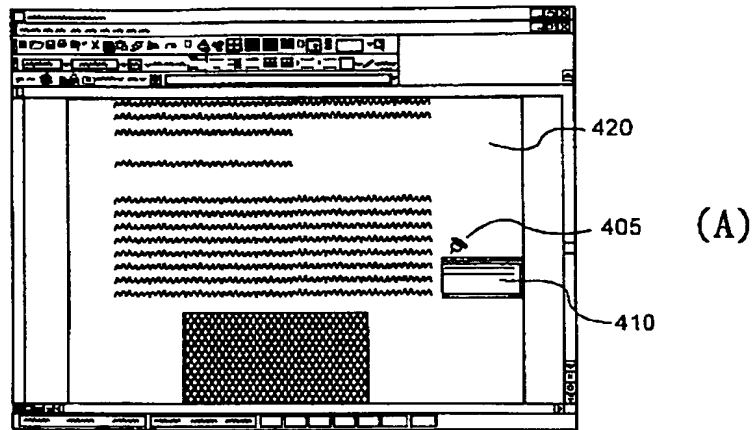
【図 3】



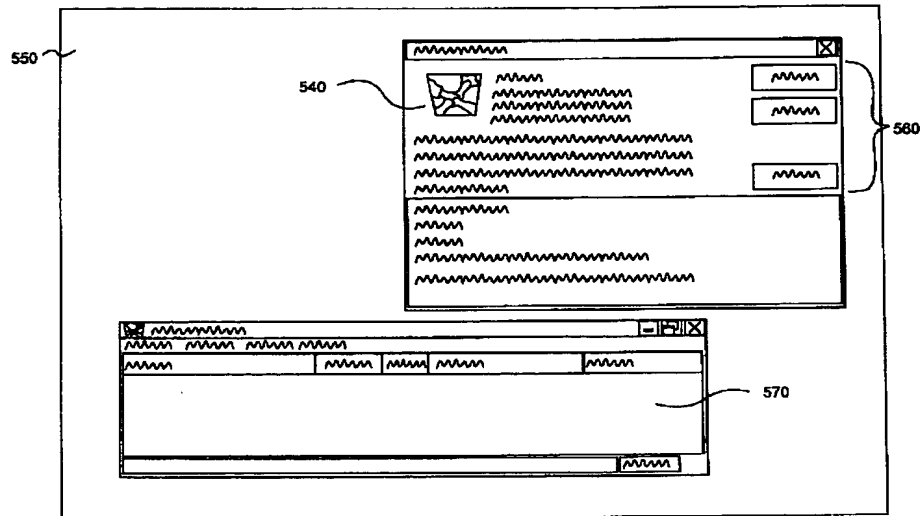
【図 5】



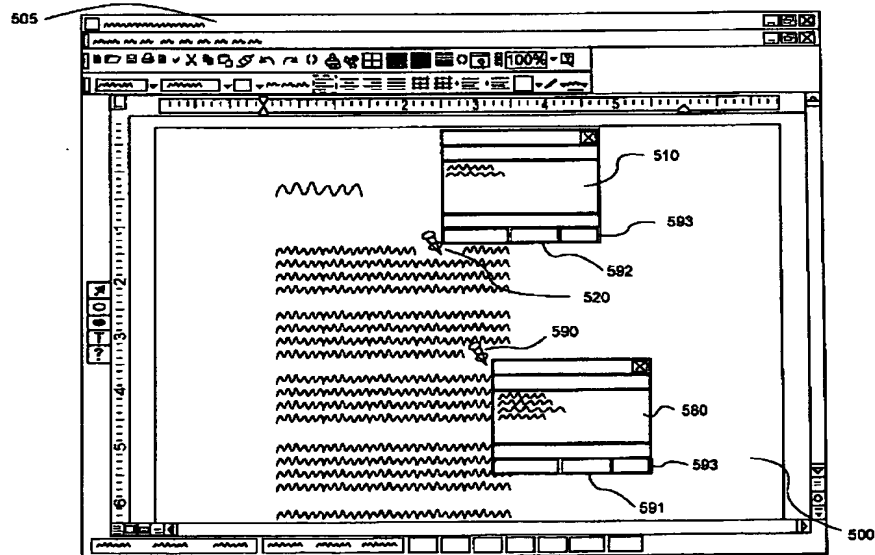
【図 4】



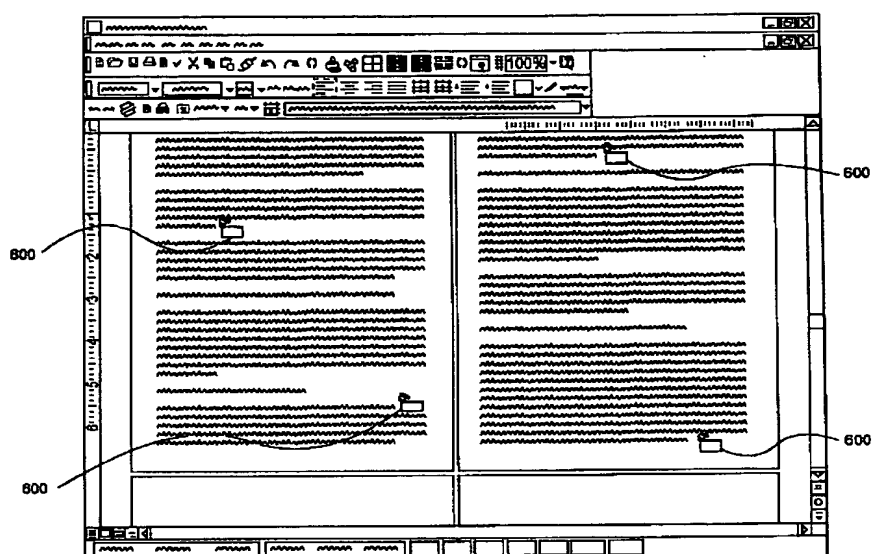
【図 6】



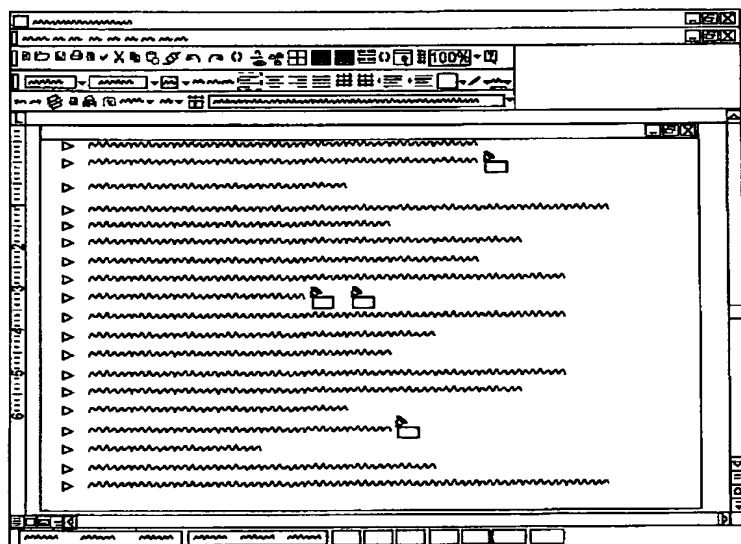
【図 7】



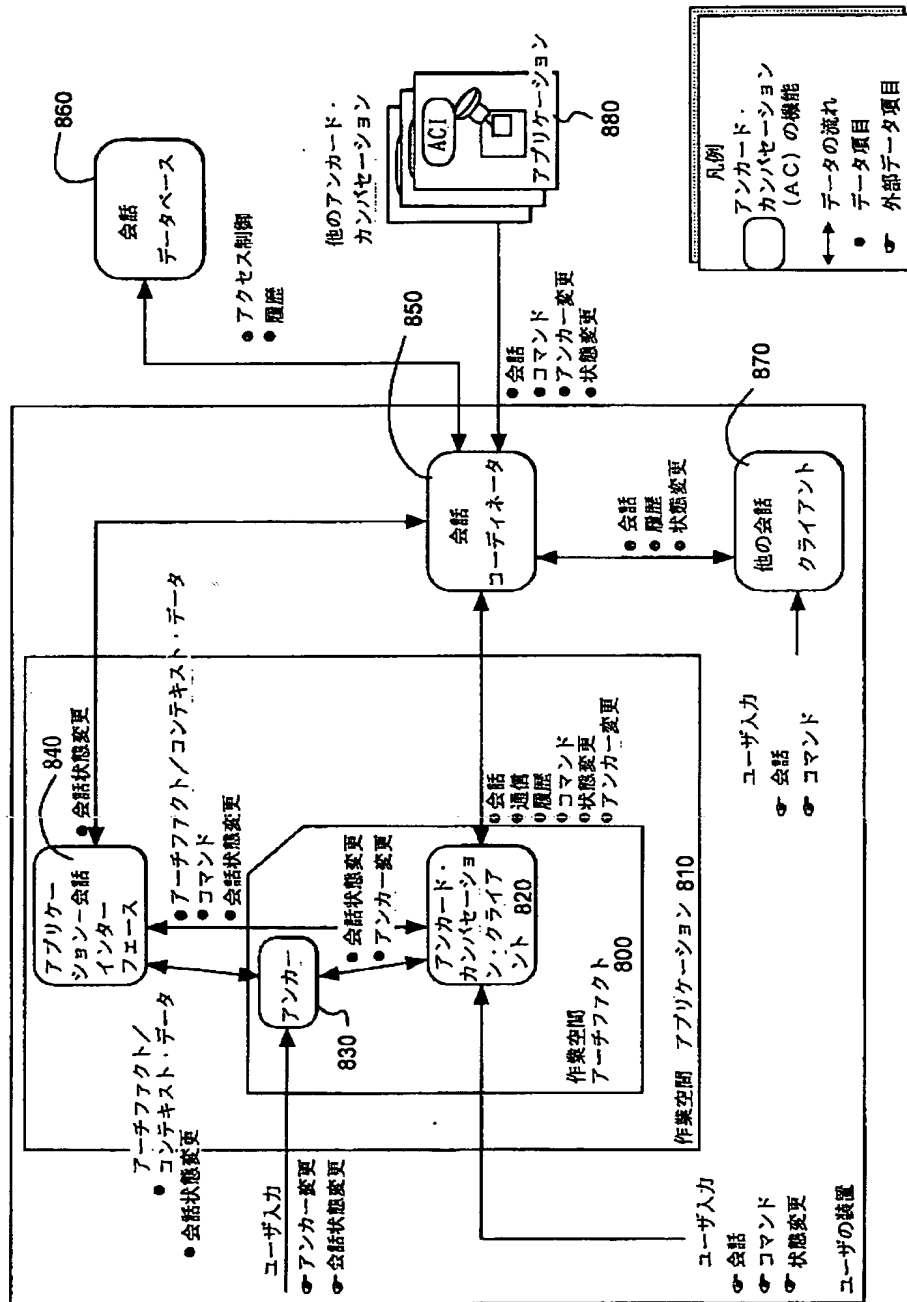
【図 8】



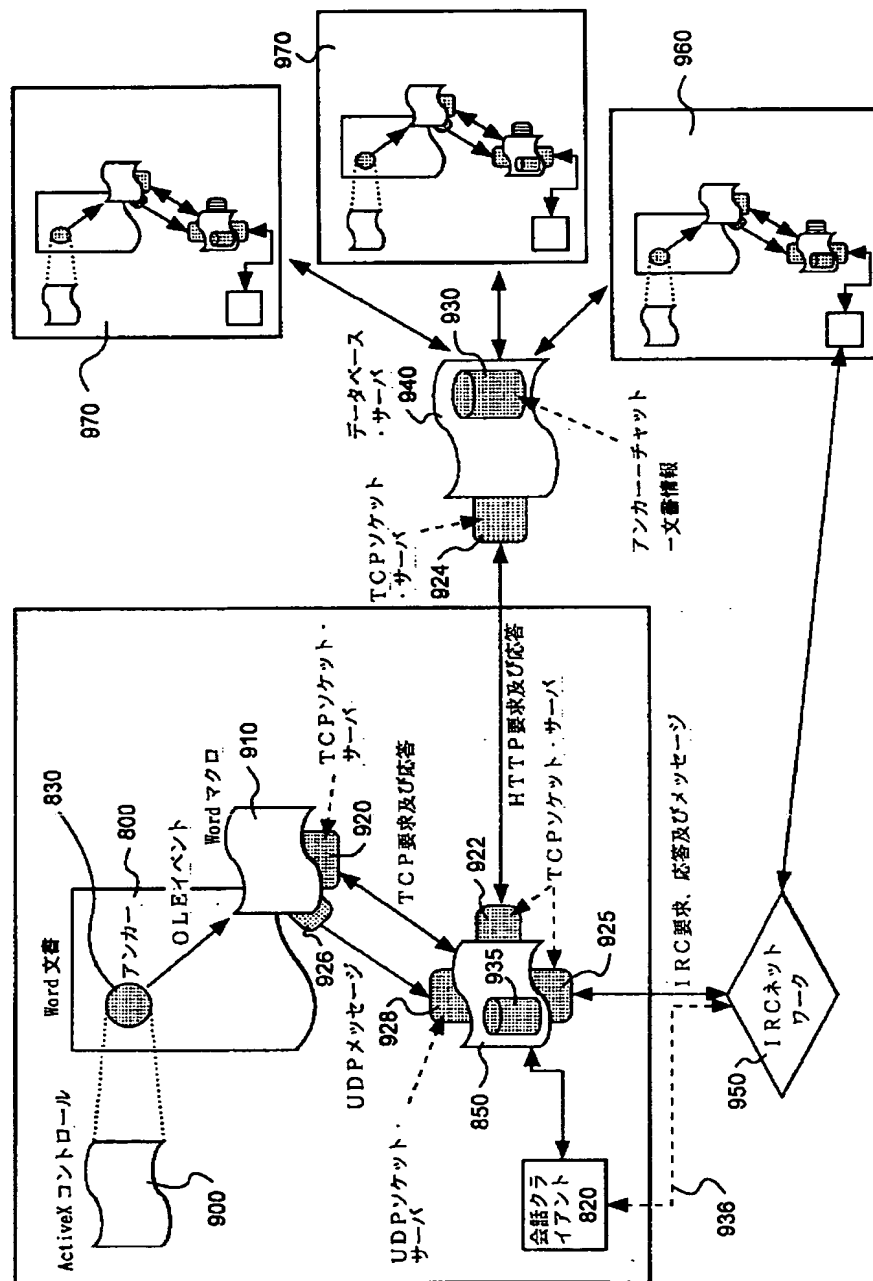
【図 9】



【図10】



【図11】



フロントページの続き

(72)発明者 レスター ディー、 ネルソン
 アメリカ合衆国 95050 カリフォルニア
 州 サンタ クララ ミシエル ジーン
 ウェイ 2453

(72)発明者 サラ ブライ
 アメリカ合衆国 97133 オレゴン州 ノ
 ース プレインズ ノース ウェスト モ
 アランド ロード 24511

(72)発明者 キヤサリン シー、 マーシャル
アメリカ合衆国 94114 カリフォルニア
州 サンフランシスコ カストロ ストリ
ート 856

(72)発明者 ジョナサン トレヴァー
イギリス国 エルエー 1 4 ダブリュージ
ー イングランド ランカスター ホール
デン ウェイ 2

(72)発明者 ジョセフ ダブリュー、 サリバン
アメリカ合衆国 94107 カリフォルニア
州 サンフランシスコ ブラクサム スト
リート 175 ナンバー 103